

ENDINEIA PICHLER

**METODOLOGIA DE ENSINO/ APRENDIZAGEM DE
CONCEITOS DE PROBABILIDADE E ESTATÍSTICA
ATRAVÉS DE UM SISTEMA TUTOR INTELIGENTE**

**FLORIANÓPOLIS – SC
JUNHO 2005**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Endineia Pichler

**METODOLOGIA DE ENSINO/ APRENDIZAGEM DE
CONCEITOS DE PROBABILIDADE E ESTATÍSTICA
ATRAVÉS DE UM SISTEMA TUTOR INTELIGENTE**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação

Marcelo Menezes Reis

Florianópolis, Junho de 2005

METODOLOGIA DE ENSINO/ APRENDIZAGEM DE CONCEITOS DE PROBABILIDADE E ESTATÍSTICA ATRAVÉS DE UM SISTEMA TUTOR INTELIGENTE

Endineia Pichler

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Banca Examinadora

Raul Sidnei Wazlawick, Dr. (coordenador)

Marcelo Menezes Reis, Dr. (orientador)

Dalton Francisco de Andrade, Dr.

Sílvia Modesto Nassar, Dr.

Jorge Coelho, DSc.

Este trabalho não é somente mérito meu, por isso dedico-o aos meus pais pelo apoio concedido, meus amigos que sempre estiveram ao meu lado dando-me força para prosseguir sempre e nunca desistir, aos meus irmãos pelo amor que nos une, em especial a Jaqueline Pichler (em memória) e ao meu namorado Ricardo Luiz Ludwig pela paciência e amor a mim dedicado.

Agradecimentos

Esta dissertação não é trabalho de um só autor, nenhum trabalho o é. Todas as pessoas que mesmo indiretamente, cruzaram o meu caminho e me fizeram pensar me moldando em quem sou hoje e contribuíram para a realização do mesmo. A elas, muito obrigada.

Agradeço aos professores da UFSC, que me receberam muito bem e passaram conhecimento para a realização desde. Muito obrigada a meu orientador Marcelo Menezes Reis por ter acreditado em meu trabalho e contribuído de forma valiosa para a realização do mesmo.

E finalmente agradeço à pessoa mais importante de todas a Deus, pelas oportunidades e desafios, e por estar sempre ao meu lado guiando meus passos.

“É melhor tentar e falhar, que preocupar-se e ver a vida passar. É melhor tentar, ainda que em vão, que sentar-se fazendo nada até o fim. Eu prefiro na chuva caminhar, que em dias tristes em casa me esconder. Prefiro ser feliz embora louco, que em conformidade viver...”

Martin Luther King

Sumário

1 INTRODUÇÃO	14
1.1 Motivação e Justificativas	14
1.2 Objetivos	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
1.3 Delimitações	16
1.4 Estrutura do Trabalho	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 Probabilidade	18
2.1.1 Experimento Aleatório	19
2.1.2 Espaço Amostral	20
2.1.3 Evento	21
2.1.4 A definição de probabilidade	23
2.1.5 Probabilidade Condicional	24
2.1.6 Eventos Independentes	25
2.1.7 Teorema da probabilidade Total	26
2.1.8 Teorema de Bayes	27
2.2 Variáveis Aleatórias	27
2.2.1 Variáveis Aleatórias Discretas	28
2.2.1.1 - Distribuição binomial	29
2.2.1.2 Distribuição de Poisson	29
2.2.2 Variável Aleatória Contínua	30
2.2.2.1 - Distribuição uniforme	31
2.2.2.2 - Distribuição exponencial	31
2.2.2.3 - Distribuição normal	32
2.3 Metodologia de ensino de Probabilidade	33
2.4 Sistemas Tutores Inteligentes	35
2.4.1 Breve Histórico	35
2.4.2 Definições, Características e Arquitetura Básica	36
2.4.3 Componentes	40
2.4.3.1 O Modelo do Especialista	43
2.4.3.2 O Modelo do Estudante	45
2.4.3.3 O Modelo Pedagógico	48
2.4.3.4 O Modelo da Interface com o estudante	51
2.4.4 “Arcabouços” e Sistemas de Autoria	53
2.4.5 Evolução para os Ambientes Interativos de Aprendizagem	54
2.4.6 Classificação	55
2.4.7 Multimídia, Hipertexto e Hipermídia	56
2.4.8 Aspectos que devem ser considerados em Sistemas Tutores Inteligentes	57
2.4.9 - Estado da Arte: Exemplos de Sistemas Tutoriais Inteligentes	58
2.4.9.1 A arquitetura MutAntIS	60

2.5 Conclusões	62
3 METODOLOGIA	64
3.1 Pesquisa de Sistemas Tutores Inteligentes	64
3.2 Metodologia desenvolvida	64
3.3 Tecnologia Utilizada para seu desenvolvimento	66
3.3.1 Borland Delphi®	67
3.3.1.1 <i>Orientação a Objeto em Borland Delphi®</i>	67
3.3.1.2 <i>Classes</i>	68
3.3.1.3 <i>Herança (Reaproveitamento de código)</i>	72
3.3.1.4 <i>Métodos virtuais e dinâmicos. Sobreposição (override) de métodos</i>	72
3.3.1.5 <i>ODBC (Open DataBase Connectivity)</i>	74
3.3.2 Borland Delphi® sua utilização	74
3.4 Conclusões	75
4 SISTEMA	76
4.1 Modelo do sistema desenvolvido.....	76
4.2 Escopo do Trabalho	77
4.2.1 Telas do sistema.....	82
4.3 – Considerações finais	89
5 CONSIDERAÇÕES FINAIS	91
5.1 Perspectivas Futuras	92
5.2. Contribuições	93
5.3 Conclusão	94
REFERÊNCIAS BIBLIOGRÁFICAS.....	95

Lista de Figuras

Figura 1 - União	Figura 2 – Intersecção	Figura 3 - Complementar	22
Figura 4 - Distribuição Uniforme.....			31
Figura 5 - Distribuição Normal			32
Figura 6 - Domínios do STIs (ICAI).....			37
Figura 7 - Arquitetura de Sistemas Tutores Inteligentes			38
Figura 8 - Diagrama de Bloco			41
Figura 9 - Estrutura básica de um Sistema Tutorial Inteligente.....			42
Figura 10 – Arquitetura <i>MutantIS</i>			61
Figura 11 - Modelo do sistema desenvolvido.....			76
Figura 12 - Interação com o sistema			78
Figura 13 – Sistema do administrador			81
Figura 14 - Login do sistema.....			82
Figura 15 - Tela Principal administrador (professor)			83
Figura 16 - Tela Principal administrador (professor)			84
Figura 17 - Acompanhamento Professor			85
Figura 18 - Acompanhamento aluno.....			86
Figura 19 - Lições			87
Figura 21 – Exercício apresentado pelo sistema			88
Figura 22 - Resultado da verificação			89

Lista de Quadros

Quadro 1 Simplificação das categorias de softwares utilizados na educação	36
Quadro 2 Simplificação dos Sistemas.....	59
Quadro 3 Implementação de uma classe.....	70
Quadro 4 Definição de classe	71

Lista de Siglas e Abreviaturas

Mwh	Mega watts hora
CAI	Computer Aided Instruction
ITSs	Intelligent Tutoring Systems - Sistemas Tutores Inteligentes – STI's , também chamados de ICAIs ,
ILEs	Interactive/Intelligent Learning Environments – Ambientes Interativos de Aprendizagem.
IA	Inteligência Artificial
ODBC	Open DataBase Connectivity.
POO	Programação orientada a objetos
SQL	Structured Query Language

Resumo

Um software educacional possui o objetivo de contribuir no processo de aquisição de conhecimento e aprendizagem. Existem diversos tipos de softwares utilizados em Educação: *Drill and Practice* (Repetição e Prática), *Tutorial Systems* (Sistemas Tutoriais), *Computer Simulations* (Simulação), *Problem-Solving software* (softwares de Resolução de Problemas), *Tool software* (software de Ferramenta) e *Computer-managed Instruction* (Instrução gerenciada por computador), dentre os quais destacam-se os sistemas tutores inteligentes, que são aplicativos capazes de tutorar uma pessoa em um determinado domínio. Um sistema tutor inteligente é composto de quatro modelos básicos: Interface, Domínio (especialista), Aprendiz (estudante) e Tutoramento (pedagógico). Através da interação desses modelos, o sistema é capaz de fazer julgamentos sobre o que o aprendiz sabe e como ele está progredindo. Os sistemas tutores inteligentes representam uma ótima ferramenta como ambientes de aprendizagem. Entretanto, tais sistemas ainda são pouco utilizados devido ao seu alto custo e longo tempo de desenvolvimento. Contudo, verificou-se a oportunidade de elaboração deste trabalho, em que o desenvolvimento de uma metodologia de ensino que resultará em um sistema tutor inteligente para auxílio ao aprendizado de alguns conceitos de probabilidade e estatística (Probabilidade, Variáveis Aleatórias Discretas e Variáveis Aleatórias Contínuas). O sistema apresenta o conteúdo e faz o acompanhamento das lições já acessadas pelo aprendiz, para que o professor possa fazer um acompanhamento do desenvolvimento de cada aprendiz. O tutor também faz a correção dos exercícios de cada lição. O objetivo do sistema é ser um auxiliar no processo de ensino, buscando sempre colaborar tanto com o professor quanto com o aprendiz.

Palavras-chave: **Software Educacional, Sistemas Tutores Inteligentes, Probabilidade, Estatística.**

Abstract

Educational software possesses the objective to contribute in the process of knowledge acquisition and learning. There are several types of educational software, amongst which the Intelligent Tutoring Systems are distinguished, capable of tutoring a person in a definite domain. An Intelligent Tutoring System is composed of four basic models: Interface, Domain (specialist), Apprentice (student) and Tutor (pedagogical). By the interaction of these models, the system can make judgments on what the current apprentice knowledge and how is its progress. Intelligent Tutorial Systems represent an excellent tool as learning environments. However, these systems are still little used because of its high financial cost and long time of development. Nevertheless, the main opportunity for this work was identified: the development of an educational methodology that will result in an intelligent tutorial system for supporting the learning of some concepts of probability and statistics (Probability, Discrete Random Variables, and Continuous Random Variables). The system presents the content and makes the accompaniment of the lesson already studied by the apprentice, allowing the teacher to evaluate the development of each apprentice. The tutor also corrects the exercises of each lesson. The objective of the system is to be a tool in the education process, looking forward to collaborate with the teacher as much as with the apprentice.

1 INTRODUÇÃO

1.1 Motivação e Justificativas

“O ensino e aprendizagem dos conceitos de probabilidade constituem um desafio para professores e estudantes. Para os estudantes é muitas vezes difícil aprender como raciocinar probabilisticamente, dando margem para a simples memorização de algumas regras que possibilitem a aprovação na disciplina que estão cursando. Os professores frustram-se ao não conseguir motivar suficientemente os alunos, e ao observar que não houve a interiorização dos conceitos. Por esse e outros motivos a maneira de ensinar está sofrendo algumas transformações, os educadores estão deixando de ser apenas transmissores de um conhecimento e passam a compartilhar, trocar e construir para e com o aluno tornando-se assim parte do processo de construção desse conhecimento juntamente com o aluno”. (Noal, 2002). E nesse processo de transformação da forma de ensinar o computador tem demonstrado ser um auxiliar na transmissão do conhecimento, através de processos pedagógicos implementados em *softwares* educacionais. Um *software* educacional possui o objetivo de auxiliar o professor no processo de ensino-aprendizagem (Azevedo, 1997), liberando-o para dedicar mais tempo ao próprio aluno e observar o progresso da aprendizagem.

O processo de ensino/aprendizagem, nesse contexto, passa a combinar objetivos sociais (interação, estratégias, etc.), afetivos (motivação, atitudes e personalidades) e cognitivos (inteligência, memória, atenção, percepção dentre outros) a fim de ampliar seus próprios horizontes teóricos (Sternfeld, 1996).

Existem diversos tipos de *softwares* educacionais, dentre os quais os sistemas tutores, que são aplicativos capazes de tutorar uma pessoa em um determinado domínio.

Os sistemas tutores constituem um ótimo recurso para auxiliar um professor em uma disciplina, para ajudar os aprendizes em seu aprendizado, e para realizar treinamentos em uma

determinada área de conhecimento, incluindo Estatística. Um passo lógico seria o desenvolvimento de aplicativos para auxílio ao ensino de probabilidade. Infelizmente tais aplicativos não são totalmente satisfatórios, pelos seguintes motivos:

- Alguns apresentam somente simulações de resultados, não explicando como resolver as questões.
- Outros são implementados em outras línguas que não o português, o que dificulta sua utilização pelos estudantes brasileiros.
- Outros ainda não incluem todos os conceitos necessários de probabilidade (deixam de lado os axiomas de probabilidade, ou variáveis aleatórias, ou distribuições de probabilidade).

Insere-se nessa deficiência a proposta deste trabalho que é o desenvolvimento de uma metodologia de ensino que incluirá um sistema tutor para auxílio ao aprendizado de conceitos de probabilidade e estatística.

1.2 Objetivos

1.2.1 Objetivo Geral

Elaborar uma metodologia que inclui sistema tutor para auxílio no ensino/aprendizagem de conceitos de probabilidade e estatística (Probabilidade, Variáveis Aleatórias Discretas e Variáveis Aleatórias Contínuas), capaz de tutorar o aprendiz em um determinado domínio.

1.2.2 Objetivos Específicos

- Investigar os conceitos de probabilidade e estatística para inclusão no sistema;

- Investigar as tecnologias de sistemas tutores inteligentes.
- Aplicar uma metodologia de ensino de probabilidade e estatística.
- Desenvolver um sistema tutorial inteligente integrado à metodologia de ensino.

1.3 Delimitações

Inicialmente havia apenas a idéia de pesquisar e elaborar uma metodologia para o ensino da probabilidade e estatística implementada em um sistema tutor inteligente. Com a estruturação do trabalho optou-se por delimitar quais conceitos seriam abordados e os conceitos escolhidos foram: probabilidade (aspectos básicos, probabilidade condicional, independência), variáveis aleatórias discretas e contínuas (os principais modelos de cada tipo). Contudo, o algoritmo básico está definido, possibilitando a adição dos outros conceitos quando necessário.

1.4 Estrutura do Trabalho

Este trabalho está dividido em cinco capítulos, dos quais este é o primeiro, onde está uma introdução do assunto, a justificativa do seu desenvolvimento e os objetivos pretendidos com este trabalho.

O Capítulo 2 traz a fundamentação teórica sobre o assunto, com os conceitos utilizados no projeto e desenvolvimento do sistema. O capítulo inicia com conceitos de Probabilidade e Estatística, como embasamento aos conteúdos para o módulo especialista desenvolvido. O capítulo é finalizado com uma abordagem generalista e desdobra-se na apresentação de definições e características pertinentes a um sistema tutor inteligente, bem como o estado da arte em sistemas já desenvolvidos.

O Capítulo 3 aponta aspectos de desenvolvimento, tomada de decisões e a descrição da metodologia desenvolvida.

No Capítulo 4 está a descrição da solução proposta, com exposição do sistema desenvolvido.

No Capítulo 5 são apresentadas as conclusões e as perspectivas para trabalhos futuros desenvolvidos a partir deste.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo será apresentada a fundamentação teórica do trabalho. Os conceitos abordados incluem as definições básicas de probabilidade (incluindo experimento aleatório, espaço amostral e eventos), variáveis aleatórias e os modelos probabilísticos mais empregados na prática. Tais conceitos constituem o âmago de qualquer curso de Probabilidade, de modo que seu exame é necessário para o desenvolvimento deste trabalho. Há também uma descrição sobre as metodologias de ensino de probabilidade, e sobre os sistemas tutoriais inteligentes, com exemplos das principais arquiteturas e aplicações desenvolvidas.

2.1 Probabilidade

A Probabilidade pode ser associada à nossa incerteza sobre o comportamento de um fenômeno qualquer. Podemos classificar os modelos matemáticos de duas formas: os modelos determinísticos onde se pode prever os resultados de um fenômeno e os modelos não determinísticos ou probabilísticos onde não se pode prever antecipadamente o resultado.

Segundo Barbeta, Reis e Bornia (2004 p.45):

“Os modelos probabilísticos são aplicados em situações que envolvem algum tipo de incerteza ou variabilidade. Mais especificamente, consideraremos a presença de algum experimento aleatório como princípio para a construção de modelos probabilísticos...”.

A construção de um modelo probabilístico para certo fenômeno permitirá fazer previsões e, portanto orientar a tomada de decisões.

O cálculo das probabilidades pertence ao campo da Matemática, entretanto a maioria dos fenômenos de que trata a Estatística são de natureza aleatória ou probabilística.

A probabilidade tem um papel muito importante nas tomadas de decisões. Existem inúmeras situações em que a probabilidade pode ajudar a determinar que caminho seguir. Os modelos probabilísticos podem ser úteis em diversas áreas do conhecimento humano (Martins e Donaire, 1993), o que justifica o seu aprendizado por profissionais tão díspares quanto biólogos, engenheiros, sociólogos e contadores.

Há alguns conceitos que precisam ser abordados antes de definir formalmente Probabilidade: experimento aleatório, espaço amostral e eventos. Seu conhecimento é crucial para a aplicação e interpretação dos resultados dos modelos probabilísticos

2.1.1 Experimento Aleatório

Ao se lançar um dado, não podemos prever qual o número que irá ocorrer. Da mesma forma, quando lançamos uma moeda não podemos prever se ocorrerá cara ou coroa. Experimentos desse tipo são chamados aleatórios. Um experimento aleatório pode ser repetido sob condições essencialmente idênticas, porém conduzindo a resultados diferentes em tentativas diferentes. Quando se diz ser possível repetir um experimento sob condições essencialmente idênticas, naturalmente está-se pensando no controle de um certo número de fatores. É claro que seria impossível controlar absolutamente todos os fatores em questão.

Na realidade, são justamente esses fatores não controlados (também chamados de variáveis de confusão, variáveis estranhas ou variáveis espúrias) que irão constituir a aleatoriedade do fenômeno. Segundo Martins e Donaire (1993 p.184);

“Experimento Aleatório é aquele que poderá se repetido sob as mesmas condições indefinidamente. Tal experimento apresenta variações de resultados, não sendo possível afirmar *a priori* qual será sua determinação antes que o mesmo tenha sido

realizado. É possível, porém, descrever todos os possíveis resultados – as possibilidades...”.

Além disso, se o experimento aleatório for repetido um número suficientemente grande de vezes ele apresentará uma *regularidade*, a qual permitirá obter as probabilidades dos diferentes resultados.

Alguns exemplos de experimentos aleatórios:

- O lançamento de um dado e a observação da face voltada para cima; não pode ser determinado a priori qual das faces vai ocorrer.
- O número de chamadas telefônicas que chegam a uma central em um determinado intervalo de tempo não pode ser determinado de antemão;
- A quantidade exata, em mm, de chuva que irá cair em uma determinada região durante um mês; o valor não pode ser determinado a priori.
- O consumo de energia, em Mwh, em uma cidade brasileira em um mês qualquer; diversos fatores podem afetar o consumo, o que impede a obtenção prévia do seu valor.

O conjunto de resultados possíveis associados a um experimento aleatório recebe uma denominação especial.

2.1.2 Espaço Amostral

Cada experimento aleatório possui um conjunto de resultados possíveis de ocorrer e esse conjunto é chamado de espaço amostral (Martins e Donaire, 1993).

Usualmente espaço amostral é denotado pela letra grega Ω .

No lançamento de uma moeda, os resultados possíveis são cara (aqui chamado K) e coroa (aqui chamado C). O espaço amostral é: $\Omega = \{K, C\}$.

No lançamento de um dado, o espaço amostral é: $\Omega = \{1, 2, 3, 4, 5, 6\}$

No lançamento de duas moedas, uma após a outra, o espaço amostral é.

$\Omega = \{(K, K), (K, C), (C, K), (C, C)\}$.

Segundo Barbetta, Reis e Bornia (2004 p.45).

“Um espaço amostral é dito discreto quando ele for finito ou infinito enumerável; é dito contínuo quando for infinito, formando por intervalos de números reais”.

A identificação da amplitude do espaço amostral, se finito ou não, é importante para a definição do modelo probabilístico que será utilizado para descrever o experimento aleatório.

Geralmente a definição do espaço amostral é o primeiro passo para obtermos a probabilidade de ocorrência de um dos seus subconjuntos, que seria o evento de interesse.

2.1.3 Evento

Segundo Barbetta, Reis e Bornia (2004 p.45).

“Chamamos de evento a qualquer subconjunto do espaço amostral:

A é um evento $\Leftrightarrow A \subseteq \Omega$ “

Um evento ocorre quando um dos resultados que o compõem ocorre. Tomando como exemplo o lançamento de um dado o espaço amostral desse experimento será o conjunto $\Omega =$

$\{1, 2, 3, 4, 5, 6\}$ Seja o evento A: sair um numero par. A conseqüentemente será um subconjunto do espaço amostral $A = \{2, 4, 6\}$.

Por causa desta definição, podemos aplicar todos os conceitos da teoria de conjuntos a eventos. Assim podemos definir eventos união, intersecção, complementar e mutuamente exclusivos, o que pode simplificar substancialmente o processo de cálculo de probabilidades. As principais operações são descritas a seguir:

- **União:** A reunião de dois eventos A e B, denotada por $A \cup B$, é o evento que ocorre se pelo menos um deles ocorre. (ver figura 1)
- **Intersecção:** A intersecção de dois eventos A e B, denotada por $A \cap B$, é o evento que ocorre se ambos os eventos ocorrerem. (ver figura 2)
- **Complementar:** Formado pelos elementos que não estão em A, é o evento que ocorre quando o evento A não ocorre (“não A”). (ver figura 3)
- **Mutuamente exclusivos:** Dizemos que os eventos A e B são mutuamente exclusivos, se eles não podem ocorrer simultaneamente, ou seja, $A \cap B = \emptyset$.

Como os eventos são subconjuntos do espaço amostral, podemos representar a união, a intersecção e o complementar de um evento pelos diagramas de Venn, tal como nas figuras 1, 2 e 3:

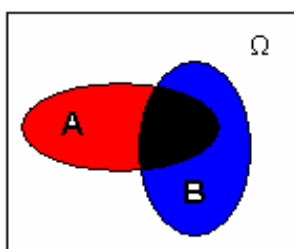


Figura 1- União

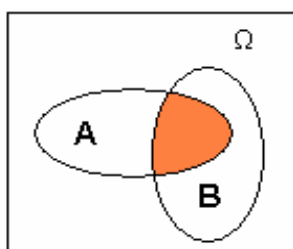


Figura 2 – Intersecção

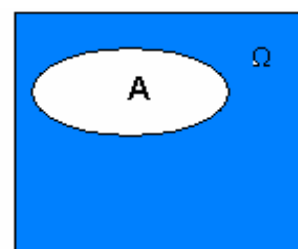


Figura 3 - Complementar

Em última análise, calculam-se as probabilidades de ocorrência de determinados eventos de interesse do espaço amostral. Recorre-se à teoria de conjuntos para representar os eventos da melhor maneira possível, facilitando as operações.

Pode-se passar agora à definição formal de Probabilidade.

2.1.4 A definição de probabilidade

Segundo Aljocyr (1976 p. 80);

“Entendendo-se por um fato (um acontecimento, um sucesso) um dos aspectos que pode ocorrer em uma experiência, podemos dizer que determinar a probabilidade de ocorrência desse aspecto é associar um número às probabilidades de que ele ocorra”.

Todas as pessoas tomam decisões levando em consideração a probabilidade de um determinado evento ocorrer. A primeira definição matemática formal da probabilidade de um evento foi baseada em simetria, ou seja, a probabilidade de um evento A ocorrer e a razão entre o número de resultados favoráveis ao evento e o número de resultados possíveis, desde que todos os resultados fossem equiprováveis:

$$\text{Probabilidade} = \frac{\text{Nº de Casos Favoráveis}}{\text{Nº de Casos Possíveis}}$$

Esta definição é algumas vezes chamada de “clássica”.

Algumas propriedades também permitem uma caracterização da probabilidade:

- Se A é o evento certo, então $P(A) = P(E) = 1$.
- Se A é o evento impossível, $P(A) = P(\emptyset) = 0$.
- A probabilidade de um evento A deve ser um número real maior ou igual a zero e menor ou igual a 1, então $0 \leq P(A) \leq 1$
- Regra da soma das Probabilidades: sejam A e B eventos quaisquer,

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Se os resultados não são equiprováveis é necessário utilizar a frequência relativa de ocorrência de um evento de interesse A:

$$f_{RA} = \frac{n_A}{n}$$

Onde n_A é o número de resultados relacionados a A, e n é o número total de realizações do experimento aleatório. Aplicando limite à frequência relativa de A, fazendo n tender ao infinito, obtém-se a probabilidade de ocorrência de A:

$$P(A) = \lim_{n \rightarrow \infty} f_{RA} = \lim_{n \rightarrow \infty} \frac{n_A}{n}$$

A definição acima é algumas vezes chamada de “experimental”, pois exigiria teoricamente a execução consecutiva do experimento aleatório.

Grosso modo, a construção de um modelo probabilístico de experimento aleatório consiste em definir o espaço amostral e obter as probabilidades de ocorrência dos eventos.

Até o presente momento foram analisados os casos em que apenas havia interesse na probabilidade de um evento individual. Mas é bastante comum querer saber qual será a probabilidade de ocorrência de um evento, supondo que outro possa ocorrer previamente: por exemplo, qual a probabilidade de chover amanhã em SC, supondo que hoje chova. Em suma, a probabilidade de ocorrência de um evento condicionada à ocorrência de outro.

2.1.5 Probabilidade Condicional

Dois ou mais eventos podem ser dependentes: a probabilidade de ocorrência de um deles é modificada pela ocorrência ou não de outros. Por exemplo, se chover amanhã a probabilidade de praticar um esporte ao ar livre será drasticamente reduzida: a ocorrência do evento “chuva” (ou a sua não ocorrência) condiciona a probabilidade do evento “prática de esporte”. Para calcular tal probabilidade deve-se usar a probabilidade condicional.

Supondo dois eventos A e B, imagine-se que a probabilidade da ocorrência de B dependa da ocorrência de A. Neste caso, a probabilidade de B será condicionada a A, que se indica por $P(B/A)$ (Martins e Donaire, 1993).

Formalmente:
$$P(B/A) = \frac{P(A \cap B)}{P(A)}$$

Segundo Barbetta, Reis, e Bornia (2004 p.52):

“Uma das conseqüências da expressão da probabilidade condicional é a regra do produto, obtida ao isolar a probabilidade de interseção”.

Se A e B **não são** independentes:

$$P(A \cap B) = P(A) \times P(B/A) \quad \text{ou} \quad P(A \cap B) = P(B) \times P(A/B)$$

É importante que seja observada com cuidado a seqüência lógica dos eventos para montar as expressões acima: na expressão da esquerda supõe-se que A ocorreu primeiro, e na da direita que B ocorreu antes.

2.1.6 Eventos Independentes

Segundo Aljocyr (1976 p. 85);

“Dois eventos serão considerados independentes, se a probabilidade de ocorrência de um deles não influenciar a probabilidade de ocorrência do outro”.

Se A e B **são** independentes (a ocorrência de um não influencia a probabilidade de ocorrência do outro):

$$P(A \cap B) = P(A) \times P(B) \quad \text{ou} \quad P(A \cap B) = P(B) \times P(A)$$

Ou seja, $P(B/A) = P(B)$ e $P(A/B) = P(A)$.

A constatação de independência pode simplificar substancialmente os cálculos de probabilidade, e constitui uma exigência para aplicação de algumas propriedades de variáveis aleatórias que serão vistas posteriormente.

É bastante usual desejar obter a probabilidade de um evento formado pela composição de outros, mutuamente exclusivos, mas de forma diferente do que foi visto até aqui. Para tal situação há o teorema da probabilidade total.

2.1.7 Teorema da probabilidade Total

Seja um evento F qualquer, referente ao espaço amostral Ω . Então:

$$F = (F \cap E_1) \cup (F \cap E_2) \cup \dots \cup (F \cap E_n)$$

onde todos os eventos $(F \cap E_i)$, são mutuamente exclusivos entre si. Logo.

$$P(F) = P\{(F \cap E_1) \cup (F \cap E_2) \cup \dots \cup (F \cap E_n)\} = P(F \cap E_1) + P(F \cap E_2) + \dots + P(F \cap E_n)$$

Usando a regra do produto, temos a seguinte equação, conhecida como o teorema da probabilidade total:

$$P(F) = P(E_1) \times P(F/E_1) + P(E_2) \times P(F/E_2) + \dots + P(E_n) \times P(F/E_n)$$

Naturalmente, algumas $P(F/E_i)$ poderão assumir valor zero por não haver interseção entre F e E_i (Barbetta, Reis e Bornia, 2004). Mesmo assim o princípio continua válido. Além disso, também pode haver interesse em obter a probabilidade de que o evento F tenha sido causado por um dos eventos individuais especificamente, o que leva à utilização do teorema de Bayes.

2.1.8 Teorema de Bayes

O teorema de Bayes é usado na inferência estatística para atualizar estimativas da probabilidade de que diferentes hipóteses sejam verdadeiras, baseadas nas observações e no conhecimento de como essas observações se relacionam com as hipóteses.

Segundo Barbetta, Reis, e Bornia. (2004 p.52)

“O teorema de Bayes está intimamente relacionado ao teorema da probabilidade total. Supõem-se as mesmas condições (eventos E_i mutuamente exclusivos e exaustivos e um evento F qualquer). Basicamente, o teorema de Bayes permite obter a probabilidade de que um dos eventos E_i ocorra, sabendo-se que o evento F ocorreu”.

Pela definição de probabilidade Condicional, temos:

$$P(E_i / F) = \frac{P(E_i) \times P(F / E_i)}{P(F)}$$

O teorema de Bayes é a base da estatística bayesiana, da qual as redes bayesianas são uma das principais aplicações.

Os conceitos básicos de probabilidade foram abordados, mas comumente os modelos probabilísticos são construídos para variáveis aleatórias, tema da próxima seção.

2.2 Variáveis Aleatórias

A observação dos resultados de um experimento aleatório é sempre um processo de medição onde uma série de mecanismos probabilísticos influencia os valores que são registrados. Nesse contexto, uma variável aleatória é aquilo que se obtém quando se associa a

cada ponto do espaço amostral um valor numérico pertencente ao conjunto dos Números Reais.

Se Ω é o espaço amostral de um experimento aleatório, e a cada resultado desse experimento associa-se uma probabilidade, a função assim estabelecida será denominada uma variável aleatória (Aljocyr; 1976). Sendo uma função, toda variável aleatória tem domínio e contradomínio: sendo o domínio o espaço amostral associado ao experimento aleatório que se está estudando, e o contradomínio o conjunto de valores que a variável pode assumir. Se o espaço amostral for finito ou infinito numerável¹ o contradomínio também o será, e a variável aleatória será dita *discreta*. Se o espaço amostral for infinito o contradomínio terá a mesma característica e a variável aleatória será dita *contínua*. O modelo então *distribui* a probabilidade do espaço amostral (que vale 1) pelos diversos valores da variável aleatória, o que justifica a denominação *distribuição de probabilidades*. Suas principais características serão discretas nas seções seguintes:

2.2.1 Variáveis Aleatórias Discretas

Segundo Aljocyr (1976 p. 85);

“Variável aleatória discreta é aquela que pode assumir determinados valores em um intervalo (conjunto), de forma que a soma das probabilidades a eles associadas seja igual à unidade”.

Quando uma variável aleatória discreta é definida, em geral sabemos quais os resultados possíveis e a probabilidade de cada um deles ocorrer. A distribuição de uma variável aleatória X é o conjunto de probabilidades associadas aos possíveis valores de X .

¹ Não se sabe exatamente quantos resultados há no espaço amostral, mas conhece-se sua lei de formação. Não se sabe qual é o número máximo de acidentes que podem ocorrer em um cruzamento, mas sabe-se que o mínimo é 0, e que o incremento é de 1 em 1: 0, 1, 2, ... acidentes.

Usualmente as variáveis aleatórias discretas são usadas para descrever processos em que há contagens (de resultados positivos dentre n tentativas, de pessoas que chegam em uma fila por hora, de peças defeituosas em um lote).

Em algumas situações não se faz necessário à construção de modelos para as variáveis aleatórias, pois existem diversos disponíveis, tornando necessário apenas definir qual deles é o mais adequado para determinada situação, simplificando a resolução do problema em questão. Os dois modelos para variáveis aleatórias mais utilizados são o binomial (distribuição binomial) e o de Poisson (distribuição de Poisson).

2.2.1.1 - Distribuição binomial

Imaginando um experimento aleatório em que pode haver apenas dois resultados possíveis, mutuamente exclusivos, “sucesso” ou “fracasso”: um experimento de Bernoulli. Se este experimento for repetido n vezes (n conhecido e finito), e em cada realização a probabilidade de sucesso p puder ser considerada constante, a distribuição da variável aleatória X, número de “sucessos” em n tentativas pode ser considerada binomial (Barbetta, Reis e Bornia, 2004).

Se houvesse interesse em calcular a probabilidade de ocorrência de s “sucessos” em n tentativas, sendo q a probabilidade de “fracasso” (1-p), tem-se:

$$P(X = s) = C_n^s \times p^s \times q^{n-s}$$

C_n^s é o número de possíveis ocorrências de s “sucessos” dentre as n tentativas.

2.2.1.2 Distribuição de Poisson

Suponha uma variável aleatória discreta X com contradomínio infinito numerável, e as seguintes condições forem satisfeitas:

- avalia-se a ocorrência dos eventos em um período contínuo (de tempo, área, volume);

- as ocorrências do evento podem ser consideradas independentes;

Neste caso a variável X segue uma distribuição de Poisson (Barbetta, Reis e Bornia, 2004). Usualmente dados históricos permitem obter uma taxa de ocorrência λ dos eventos no período considerado, o que possibilita calcular a probabilidade de ocorrência de k eventos:

$$P(X = k) = \frac{e^{-\lambda} \times \lambda^k}{k!}$$

A distribuição de Poisson é aplicável em fenômenos nos quais eventos aleatórios ocorrem através do tempo tais como acidentes, ligações falsas em sistemas telefônicos, etc.

Há vários outros modelos probabilísticos para variáveis aleatórias discretas, tais como hipergeométrico, geométrico, de Pascal.

A seguir serão abordados os principais aspectos referentes às variáveis aleatórias contínuas.

2.2.2 Variável Aleatória Contínua

Segundo Aljocyr (1976 p. 85);

“Seja o intervalo $[a;b] \in \mathbb{R}$. A variável aleatória que assumir todos os valores contidos no intervalo $[a;b] \in \mathbb{R}$ é denominada de variável aleatória contínua”.

Usualmente as variáveis aleatórias contínuas são resultado de medição, feita por algum instrumento. Exemplos de variável aleatória contínua:

- A espessura, em milímetros, de um determinado parafuso.
- O tempo de vida média, em minutos, de determinados componentes eletrônicos.

Uma variável aleatória contínua pode ser descrita por sua função densidade de probabilidades $f(x)$. Para estudar um fenômeno através de uma variável aleatória contínua é preciso definir $f(x)$. Felizmente, da mesma forma que para variáveis aleatórias discretas há vários modelos (distribuições) desenvolvidos e disponíveis para aplicação nas mais diversas situações, como os modelos uniforme, exponencial e normal, dentre muitos outros. A seguir serão apresentados alguns destes modelos.

2.2.2.1 - Distribuição uniforme

Trata-se provavelmente do mais simples modelo probabilístico para variáveis aleatórias contínuas. Dois intervalos de valores da variável aleatória contínua, que tenham o mesmo tamanho, tem a mesma probabilidade de ocorrer. Formalmente, uma variável aleatória contínua X tem distribuição uniforme, com parâmetros a e b reais (sendo a menor do que b), se sua função densidade de probabilidades for (Barbetta, Reis e Bornia, 2004):

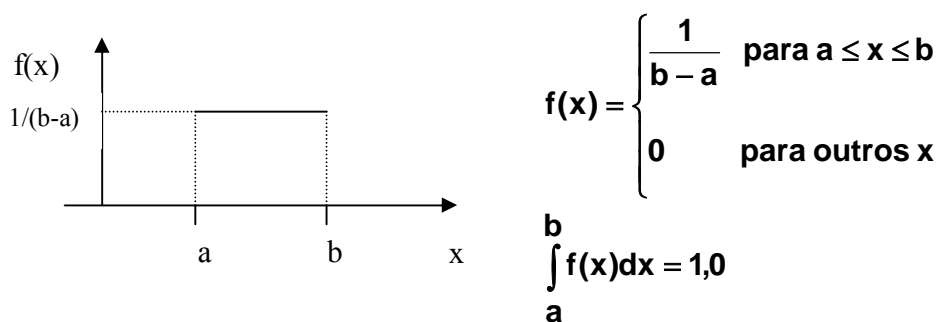


Figura 4 - Distribuição Uniforme

2.2.2.2 - Distribuição exponencial

O modelo exponencial tem uma forte relação com o modelo de Poisson. A distribuição de Poisson modela o número de ocorrências em um período contínuo (de tempo, de comprimento, de área, de volume). À **distância** entre estas ocorrências (seja medida em minutos, metros, metros quadrados) também é uma variável aleatória, mas agora contínua, que pode ser modelada pela distribuição exponencial.

Formalmente, “uma variável aleatória contínua X que é igual à distância entre contagens sucessivas de um processo que segue uma distribuição de Poisson, cuja média vale λ , segue uma distribuição exponencial com parâmetro λ ”. Sua função densidade de probabilidades será (Barbetta, Reis e Bornia, 2004):

$$f(x) = \begin{cases} \lambda \times e^{-\lambda \times x} & \text{para } x \geq 0 \\ 0 & \text{para } x < 0 \end{cases}$$

2.2.2.3 - Distribuição normal

Trata-se de uma das distribuições contínuas mais importantes: permite modelar uma infinidade de fenômenos naturais, e devido a algumas de suas características possibilita realizar aproximações para calcular probabilidades de ocorrência de valores de variáveis aleatórias que seguem outras distribuições (contínuas e discretas). É muito importante também para Inferência Estatística e Controle Estatístico da Qualidade. O gráfico da função densidade de probabilidades de uma variável aleatória contínua X que tenha distribuição normal será (Barbetta, Reis e Bornia, 2004):

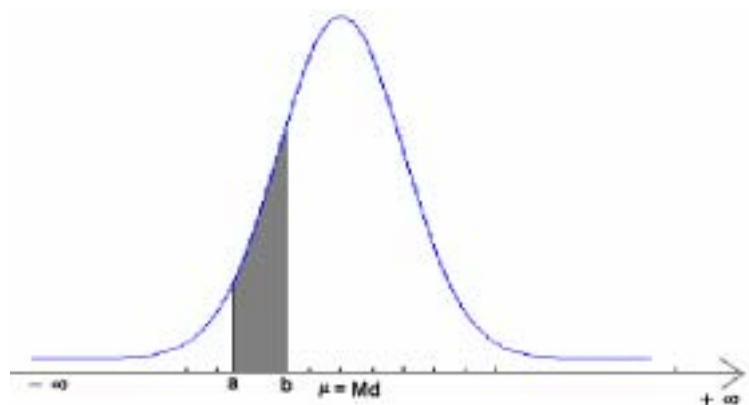


Figura 5 - Distribuição Normal

E sua função de densidade de probabilidade será:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{1}{2}\times\left[\frac{x-\mu}{\sigma}\right]^2\right)} \quad -\infty < x < +\infty$$

Para calcular a probabilidade de ocorrência de valores de X entre a e b na figura 5 basta calcular a integral de f(x) acima entre a e b: mas como a integral não tem solução definida, utilizam-se tabelas com os resultados das integrais numéricas, disponíveis em qualquer livro de Estatística.

Uma das características mais importantes da distribuição normal é a possibilidade de usá-la para aproximar outras distribuições, devido ao teorema do limite central, o que também permite o seu emprego em inferência estatística.

Passa-se agora para a discussão das metodologias de ensino de Probabilidade.

2.3 Metodologia de ensino de Probabilidade

Para muitos estudantes, Probabilidade é o tópico mais difícil de aprender em um curso de Estatística. Russel (1998) declara que mesmo estudantes com boa base matemática chegam a considerar o tópico às vezes maçante. Para os professores também é difícil, devido ao pouco tempo disponível e à complexidade do assunto, além da já citada atitude dos estudantes (Keeler e Steinhorst, 2001).

Muitos professores e pesquisadores desenvolveram técnicas para facilitar a aprendizagem e tentar estimular o raciocínio probabilístico dos seus aprendizes, nos mais variados níveis de ensino.

Bastante comum é a utilização de jogos lúdicos comuns, como dados, gamão, cartas, para explicar os conceitos de experimento aleatório e desenvolver o raciocínio probabilístico. Tal abordagem é utilizada até mesmo com crianças, tal como relatam Amit (1998) e Truran (1998). Em tais casos o instrutor, ou professor, procura motivar os aprendizes a raciocinar

sobre a probabilidade de vencer o jogo, verificar se o jogo é honesto, entre outros aspectos. Uma das desvantagens desta abordagem é a sua limitação intrínseca às “regras” dos jogos, onde muitas vezes usa-se apenas a definição clássica de probabilidade, dificultando a generalização do raciocínio para situações mais complexas. De qualquer forma, mesmo que apenas como passo introdutório à estratégia de usar jogos no ensino da probabilidade é bastante válida.

Naturalmente, com a popularização dos computadores pessoais e da Internet, vários pesquisadores desenvolveram *softwares* educacionais nas mais diversas áreas. Especificamente em Estatística diversas tentativas foram feitas:

- Usando planilhas eletrônicas comerciais, ou suplementos que podem ser acessados de dentro de tais planilhas (Boyle, 1998) (Feu, Hunt e Lees, 1998);
- Usando *software* especificamente projetado, incluindo mecanismos de simulação (Finzer e Erickson, 1998);
- Utilização de conceitos de Inteligência Artificial, como sistemas especialistas (Nassar, 1995) (Cechinel e Moreira, 1998).
- E muitas aplicações baseadas na Internet.

Chama a atenção o fato de que a maioria dos aplicativos destina-se ao ensino de Estatística, e apenas marginalmente ao ensino de Probabilidade. Nota-se também a quase ausência de sistemas tutoriais inteligentes (que serão discutidos mais à frente neste capítulo), com exceção de alguns que ou tiveram sua pesquisa encerrada, como o VITAL (Clarkson et al., 1994), ou são direcionados para um tópico específico de Estatística, como o Controle Estatístico da Qualidade (Reis, 2001).

Em muitos casos é comum a combinação de jogos, aplicativos computacionais, calculadoras e atividades em grupo (cooperativas) para a implementação do ensino. Contudo, em muitos casos o *software* educacional usado não possibilita o tutoramento do estudante fora do ambiente de sala de aula, o que um sistema tutorial inteligente sobre Probabilidade poderia propiciar.

2.4 Sistemas Tutores Inteligentes

Um *software* educacional possui o objetivo de contribuir no processo de aquisição de conhecimento e aprendizagem. Existem diversos tipos de *softwares* educacionais, dentre os quais destacam-se os sistemas tutores inteligentes, que são aplicativos capazes de tutorar uma pessoa em um determinado domínio, esta seção apresenta uma visão geral dos Sistemas Tutores Inteligentes, sua evolução histórica e os recursos de hipermídia que foram incorporados a estes sistemas fornecendo maior controle no processo de aprendizagem por parte do usuário.

2.4.1 Breve Histórico

Os primeiros programas de instrução assistida por computador (CAI – Computer Aided Instruction) foram concebidos para melhorar a performance do estudante em domínios baseados em habilidades. Estes programas tinham uma visão comportamental do processo de aprendizado. Não forneciam os resultados de como as pessoas aprendem. Assumiam que, dada uma informação ao estudante, este poderia absorvê-la Urban-Lurain (1996).

Ainda na década de 60 surgem os *Micromundos*, tendo uma proposta diametralmente oposta aos CAIs, ou seja, o aprendiz passa a construir seu próprio conhecimento. A transição para os ITSs (Intelligent Tutoring Systems - Sistemas Tutores Inteligentes – STI's), também chamados de ICAIs, se dá no início da década de 80. Nos STIs a noção de modelo é de fundamental importância: modelo do domínio, modelo do estudante e modelo do processo de comunicação. Enfatizando este aspecto da comunicação, Wenger (1987) redireciona as pesquisas dos STIs focando a parte cognitiva como preponderante em relação aos modelos computacionais pedagógicos e do domínio Urban-Lurain (1996).

Da combinação dos STIs e *Micromundos* surgem os ILEs (Interactive/Intelligent Learning Environments – Ambientes Interativos de Aprendizagem). Enquanto que nos STIs o sistema detém o controle da interação, esta passa a ser mais flexível nos ILEs.

Abaixo apresentamos um quadro que exemplifica as categorias de softwares utilizados na Educação.

Quadro 1 Simplificação das categorias de softwares utilizados na educação.

Software	Particularidade
Drill and Practice (Repetição e Prática)	Programas de exercício-e-prática são utilizados para revisar material visto em classe principalmente, material que envolve memorização e repetição, como aritmética e vocabulário.
Tutorial Systems (Sistemas Tutoriais)	Sistemas Tutores Inteligentes são sistemas voltados ao ensino que buscam modelar aspectos envolvidos na tutoria humana
Computer Simulations (Simulação)	Estes modelos permitem a exploração de situações fictícias. A simulação oferece a possibilidade do aprendiz desenvolver hipóteses, testá-las, analisar resultados e refinar os conceitos.
Tool software (software de Ferramenta)	O computador pode ser usado também como ferramenta educacional. Segundo esta modalidade o computador não é mais o instrumento que ensina o aprendiz, mas a ferramenta com a qual o aluno desenvolve algo, e, portanto, o aprendizado ocorre pelo fato de estar executando uma tarefa por intermédio do computador.
Computer-managed Instruction (Instrução gerenciada por computador).	Trata-se do uso do computador como elemento de apoio à aprendizagem. Refere-se a seu emprego como auxiliar em cálculos de um modo geral, levantamento bibliográfico ou ainda para sua inter-relação com outro de transmissão de dados à distância ou qualquer outro dispositivo

2.4.2 Definições, Características e Arquitetura Básica.

Sistemas Tutores Inteligentes são sistemas voltados ao ensino que buscam modelar aspectos envolvidos na tutoria humana. São referenciados na literatura como sistemas que sabem *o que ensinar (conteúdo)*, *para quem ensinar (modelagem do aprendiz)* e *como ensinar*

(*estratégias pedagógicas ou de ensino*). Métodos e técnicas de Inteligência Artificial podem ser utilizados para proporcionar uma maior adaptação do sistema ao aprendiz.

Um Sistema Tutor Inteligente (STI) é definido, segundo Auberger (1998), como um sistema instrucional baseado em computador, que ensina o estudante de uma maneira interativa, usando os conceitos de Inteligência Artificial.

Murray ressalta como um dos objetivos dos STIs, ser capaz de modelar complexos comportamentos de ensino, os quais se adaptam às necessidades do estudante, à situação de aprendizagem e ao assunto da instrução Murray (1999).

Para Kearsley, os STIs fazem parte de um campo interdisciplinar envolvendo Ciência da Computação, Psicologia cognitiva e Educação conforme representado na Figura 1 Kearsley (1987).

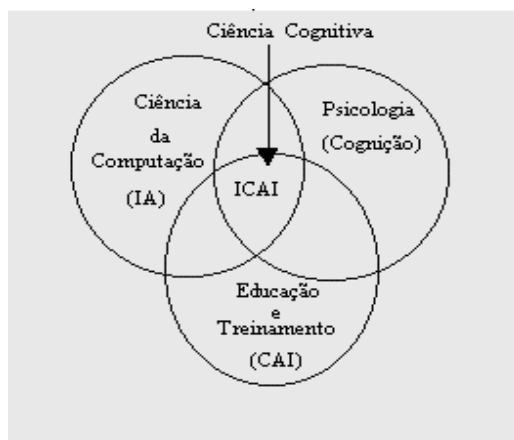


Figura 6 - Domínios do STIs (ICAI)

A arquitetura de Sistemas Tutores Inteligentes varia de uma implementação para outra. Entretanto, existem alguns componentes ou módulos básicos que podem ser observados em diversos trabalhos Wenger (1987), Yazdani (1987), Barr e Feigenbaum (1982) e na maioria dos casos sob diversas denominações, mas que possuem o mesmo propósito básico. Esses componentes básicos são demonstrados na Figura 7 Wenger (1987) Giraffa (1999).

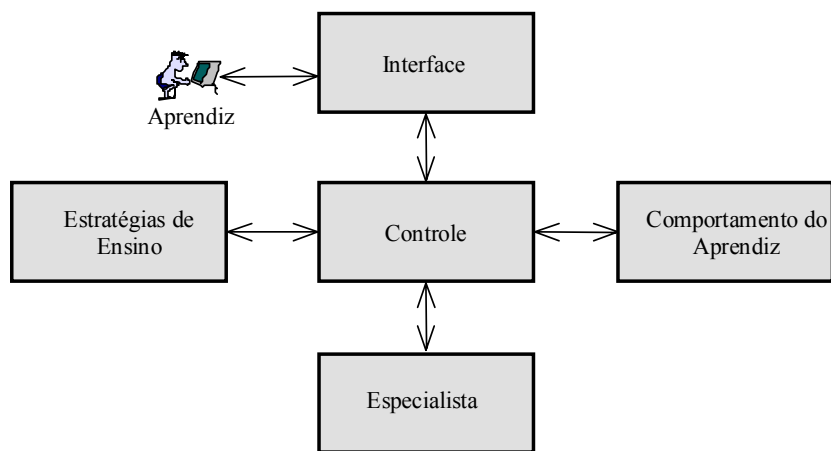


Figura 7 - Arquitetura de Sistemas Tutores Inteligentes

Especialista - O módulo especialista é responsável por manipular o conteúdo que vai ser ensinado pelo STI. Nele podem ser encontrados: o material instrucional, mecanismos de geração de exemplos, formulação de diagnósticos e os processos de simulação que podem existir nos STI's. Esse componente do sistema é muitas vezes referenciado na literatura com a denominação de “Base de Conhecimento do Domínio”. Vários modelos de representação de conhecimento podem ser utilizados para o seu desenvolvimento, tais como: redes semânticas, regras de produção e *frame*.

Um dos fatores que diferencia um STI de um CAI convencional é o fato do conteúdo ser armazenado em uma base de conhecimento, com capacidade de “raciocínio” e não em uma base de dados convencional estática, contendo todos os fatos pertencentes ao domínio.

Comportamento do Aprendiz - O módulo do comportamento do aprendiz deve captar o estado do entendimento do aprendiz a respeito do assunto que está sendo apresentado Wenger (1987). Os dados que fazem parte deste módulo são de fundamental importância para que o STI possa comprovar hipóteses a respeito do aprendiz. É interessante que esse módulo seja atualizado dinamicamente, à medida que o sistema avalia o desempenho do aprendiz.

Estratégias de Ensino - O módulo de Estratégias de Ensino, também chamado de *módulo tutor*, *módulo instrutor*, *módulo pedagógico* ou ainda *módulo de estratégias didáticas*, deve ser capaz de tomar as decisões sobre as estratégias de ensino a serem utilizadas e determinar as informações que serão apresentadas a um aprendiz, de forma semelhante a um professor. Pode-se dizer que o comportamento de um STI é determinado diretamente por este módulo.

Segundo Ohlsson (1987), o STI deve ter dois níveis de planejamento: estratégias pedagógicas e táticas pedagógicas, onde as estratégias possuem o conhecimento sobre como ensinar, ou seja, como gerar uma seqüência de táticas para apresentar com sucesso um determinado tópico a um determinado aprendiz e as táticas são as ações necessárias para exteriorizar uma estratégia. Desta forma, pode-se concluir que: uma estratégia didática particular é composta de um conjunto de táticas (ou ações) para realizá-la.

Existem diversos métodos de estratégias que podem ser utilizados em STI's. Alguns deles são os seguintes Giraffa (1999):

- Método Socrático: esse método permite que o STI ensine através de perguntas e diálogos, levando o aprendiz a tirar suas próprias conclusões.
- Método Colaborativo (Assistente): o STI comporta-se como um colaborador na interação com o aprendiz, ajudando-o a esclarecer suas idéias.
- Método *Coaching* (Treinamento): esse método faz com que o STI observe o desempenho do aprendiz, a fim de aconselhá-lo nas realizações de suas atividades.

Em outro nível, as táticas que podem ser utilizadas em estratégias de ensino são as seguintes Pereira *et al.*, (1998): mostrar um exemplo usando uma situação similar; mostrar uma mensagem com a melhor opção; mostrar exemplos relacionados sem nenhuma explicação; mostrar temas que são importantes prestar mais atenção; mostrar o conteúdo de cada tópico; mostrar uma mensagem explicando as conseqüências de suas ações; mostrar ao aprendiz sucessivas questões para que ele possa analisar hipóteses, descobrir contradições e realizar inferências corretas.

O método utilizado no STI é que deve determinar quais táticas devem ser utilizadas em uma estratégia didática. Entretanto, caso o projetista do STI não esteja adotando nenhum método específico, ele deverá escolher o conjunto de táticas para compor uma estratégia que melhor satisfazem os objetivos.

É importante ressaltar que muitos dos STI's citados na literatura têm mais de uma estratégia didática e isto ocorre porque os sistemas geralmente têm princípios diferentes de instruir, tal qual um professor humano tem diferentes maneiras de apresentar o mesmo assunto a um aprendiz.

Interface - O módulo de Interface é responsável pela comunicação entre o aprendiz e o STI. Através da Interface, o STI pode apresentar o seu material instrucional e monitorar o progresso do aprendiz pela recepção de suas respostas.

Alguns autores consideram que uma boa interface é vital para o sucesso de qualquer sistema interativo. Em STI's, pode-se dizer que a questão da interação cresce em importância, já que se deve proporcionar ao aprendiz uma interação o mais amigável possível. Por isso, muitos desenvolvedores de STI's optam pelo desenvolvimento de Interfaces Adaptativas.

Controle - O módulo de controle é responsável pela coordenação geral do STI e trata da comunicação entre os módulos, interface e eventuais chamadas a outros programas utilitários. Em alguns casos, esse módulo não é encontrado de maneira explícita nas arquiteturas e o controle fica distribuído entre os diversos módulos.

2.4.3 Componentes

Apesar de haver diversas arquiteturas para STIs, há um consenso quanto aos componentes básicos: *Módulo Especialista*, *Módulo do Estudante*, *Módulo Pedagógico (Tutor)* e *Módulo da Interface* Woolf (1988), Wenger (1987) e Kearsley (1987).

A divisão clássica de um STI em módulos resulta no diagrama de blocos apresentado na Figura 8 Kaplan, (1995). Nela distinguem-se quatro entidades básicas: o Modelo Pedagógico, o Modelo do Especialista, o Modelo do Estudante, o e a Interface.

O Modelo Pedagógico, também chamado Modelo Instrucional ou Regras de Ensino, executa o diagnóstico do conhecimento do aprendiz, decide quais as estratégias de ensino serão utilizadas e determina a maneira que a informação será apresentada.

O Modelo do Especialista, ou Rede de Conhecimento descreve o conhecimento de um especialista na área de domínio do sistema, servindo como base para a construção do Modelo do Estudante.

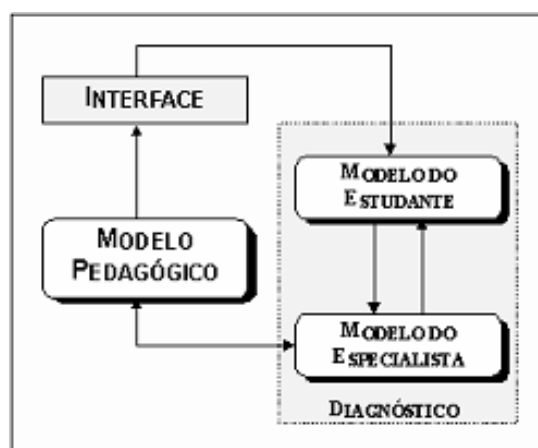


Figura 8 - Diagrama de Bloco

O Modelo do Estudante é a representação do conhecimento do aprendiz e dos seus erros ou mal-entendidos, mapeando quais informações do tutor já foram assimiladas.

É através da comparação entre as informações assimiladas pelo aprendiz (Modelo do Estudante) e o Modelo do Especialista, através de um processo convencionalmente denominado de Modelo Diferencial, que são executados os processos de diagnóstico, alimentando o Modelo Pedagógico acerca da performance do usuário em questão.

O módulo de Interface realiza o intercâmbio de informações entre o sistema, o instrutor e o aprendiz. Ele apresenta material apropriado ao nível de entendimento do aprendiz e mantém a coerência nas explicações.

O desenvolvimento de um STI requer a aplicação integrada de todos esses quatro modelos, cujas inter-relações podem ser compreendidas de acordo com a Figura 9. A modelagem da cognição humana, a representação do conhecimento do especialista, e o sistema de interface inteligente do usuário constituem somente um subconjunto das capacidades funcionais requeridas no processo de ensino/aprendizagem, pois, são únicas e muito mais complexas do que aquelas apresentadas em outros sistemas inteligentes.

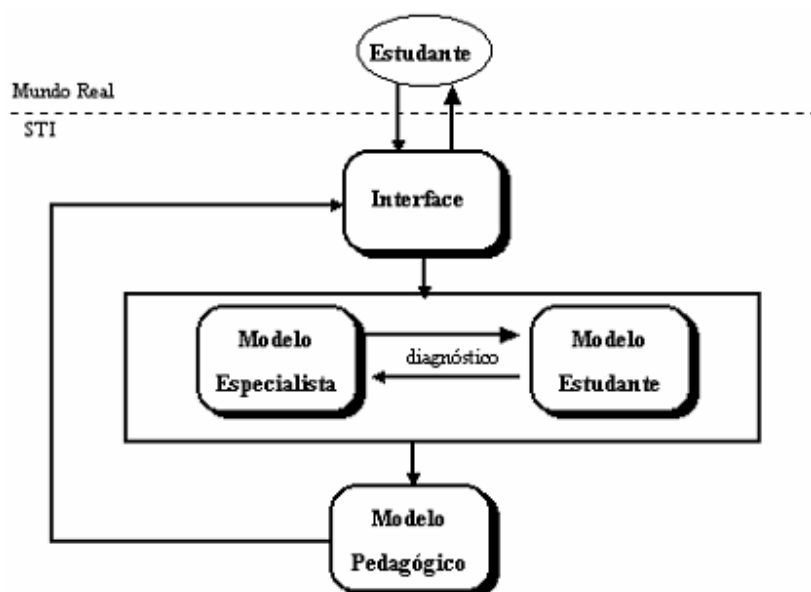


Figura 9 - Estrutura básica de um Sistema Tutorial Inteligente

O desenvolvimento de um STI requer, portanto, uma abordagem sistemática para integrar os vários tipos de especialidades dentro de um único sistema Park, (1988). Durante uma sessão educacional, o sistema monitora a performance do estudante e tenta apurar o conhecimento que o estudante detém. Este processo de diagnóstico é realizado pela comparação do estado de conhecimento atual do estudante com o conhecimento contido no

modelo do especialista. Os resultados desta comparação são passados para o modelo pedagógico, onde as decisões são tomadas sobre qual, quando e como a informação será transmitida através da interface do sistema com o estudante.

2.4.3.1 O Modelo do Especialista

O Modelo do especialista contém o conhecimento do domínio do sistema, e os mecanismos de inferência. Este modelo tem duas tarefas: serve como o conhecimento a ser apresentado, e ao mesmo tempo como um padrão para avaliar a performance do estudante Auberger, (1998).

O módulo do especialista pode ser classificado em dois tipos de arquitetura, com relação à representação: *caixa-preta* (o sistema avalia o que o estudante produz, e não como isto foi alcançado. Somente os resultados finais são disponíveis), e *transparente* (cada passo no raciocínio do sistema, pode ser inspecionado e interpretado) (Wenger, 1987).

Wenger (1987) faz uma distinção entre conhecimento compilado e conhecimento articulado. O conhecimento compilado pode ser visto como tão especializado a ponto de ter perdido a transparência e generalidade. Já o conhecimento articulado tende a incluir justificativa, podendo ser revisto dinamicamente e adaptado a novas situações.

O modelo do especialista é fundamentalmente uma base de conhecimento, contendo informações de um determinado domínio, que é organizada de alguma maneira para representar o conhecimento de um especialista ou professor. É, geralmente, considerado o componente central de qualquer STI. Em essência, este modelo incorpora a maior parte da “inteligência” do sistema na forma do conhecimento necessário para solucionar problemas do domínio (Park, 1988).

Esta base de conhecimento contém os elementos para que o estudante aprenda o conhecimento do domínio e os procedimentos necessários para que ele possa utilizá-los na

resolução dos problemas em uma determinada área deste domínio. Para isto, este conhecimento deve ser mapeado em símbolos para que o computador possa armazená-lo e manipulá-lo durante as interações com o estudante (Orey, 1993).

Na transição de CAI para STI, o conhecimento do domínio foi o primeiro aspecto da experiência do professor a ser explicitamente representado nos sistemas. Nos sistemas CAI, o conhecimento a ser transmitido estava contido em blocos de apresentação embutidos no sistema sendo apresentados aos estudantes sob determinadas condições (Wenger, 1987).

Assim, o modelo do especialista desempenha dupla função:

- Age como uma fonte para o conhecimento a ser apresentado. Isto inclui tanto a geração de explicações e respostas aos estudantes, como também tarefas e questões.
- Serve como um padrão para as avaliações de desempenho do estudante. Para esta função, deve ser capaz de gerar soluções para problemas no mesmo contexto que o estudante realiza, para que as respectivas respostas possam ser comparadas.

Várias abordagens para modelar o especialista e representar seu conhecimento têm sido exploradas nas pesquisas em STIs. A aquisição do conhecimento necessário para este modelo pode ser a principal tarefa no desenvolvimento do sistema, requerendo muitas horas de colaboração entre o projetista e o especialista ou professor. Um modelo do especialista bem projetado facilitará a comunicação do conhecimento entre o professor e o estudante. Entretanto, quanto maior esta capacidade, maior será a complexidade do sistema, e isto, na maioria das vezes, pode não ser necessário para uma instrução eficaz. Pode ser suficiente representar este conhecimento para um conjunto reduzido de problemas a serem usados com propósitos educacionais, simplificando o desenvolvimento do modelo.

O grande desafio para cada novo STI é fornecer uma rica representação do seu domínio, suficiente para suportar o nível desejado de compreensão e, portanto, proporcionar flexibilidade no ensino. A estrutura e inter-relação do conhecimento necessária para um

sistema tutorial são muito maiores do que aquelas necessárias aos típicos sistemas especialistas. Além disso, a forma na qual este conhecimento é armazenado é crucial para a capacidade do sistema em utilizá-lo. Nenhuma forma geral parece adequada para representar o conhecimento, mas tipos diferentes de raciocínio e de conhecimento requerem diferentes representações para um uso eficiente e eficaz Rickel, (1989). Portanto, a escolha da representação de conhecimento em um sistema tutorial depende do tipo de conhecimento a ser armazenado e da utilização pretendida. Alguns métodos de IA utilizados para representar o conhecimento do domínio, incluem o desenvolvimento de redes semânticas, a aplicação de regras de produção, representações procedimentais, e a construção de *frames* e *scripts*.

Várias técnicas empregadas por projetistas educacionais para identificar e organizar conteúdos podem também ser utilizadas para adquirir o conhecimento necessário para o modelo do especialista, assim como várias técnicas usadas pelos engenheiros de conhecimento podem ser úteis para projetos educacionais mais tradicionais.

A linha mestra então, é que o “comportamento inteligente” - e, portanto, a capacidade tutorial - requer um conhecimento rico e suficientemente estruturado de modo a facilitar os tipos desejados de raciocínio e as capacidades cognitivas envolvidas no processo de ensino e aprendizagem.

2.4.3.2 O Modelo do Estudante

A chave para um ensino personalizado e inteligente em um sistema tutorial é, sem dúvida, o conhecimento que o sistema deve ter de seu próprio usuário. A dimensão mais significativa de um STI é sua capacidade para modelar o conhecimento do estudante (Jonassen, 1993a).

A característica principal deste modelo deve contemplar todos os aspectos do conhecimento e do comportamento do estudante que tragam consequências para o seu desempenho e aprendizagem. Entretanto, a construção de um modelo como este é uma tarefa bastante complexa para um sistema computadorizado. Os canais de comunicação em um

computador podem parecer bastante restritos quando comparados com a capacidade das pessoas em combinar informações em uma grande variedade de meios, como, por exemplo, o tom de voz ou expressões faciais.

Mesmo que para obter decisões pedagógicas razoáveis não se tenha a necessidade de construir um modelo completo do estudante ao longo de todas as suas dimensões, a construção de um modelo parcial que forneça somente as informações necessárias é, ainda hoje, um desafio para os sistemas computacionais.

Este modelo deve ser dinâmico, contendo o conhecimento e as capacidades do estudante, seu comportamento de aprendizagem passado, os métodos de apresentação aos quais ele responde melhor e sua área de interesse dentro do domínio. Munido destas informações, o sistema pode atingir um nível desejável e um método de apresentação adequado, adaptando a instrução à competência e habilidade de cada estudante.

O processo de formação e atualização do modelo do estudante pela análise dos dados disponíveis do sistema é freqüentemente chamado de diagnóstico. Se por um lado o diagnóstico puramente inferencial - e que exclui o estudante - é uma tarefa bastante difícil, por outro lado a abordagem puramente interativa sofre pelo fato de que as pessoas nem sempre conseguem explicar seus próprios processos mentais e muitas vezes ficam totalmente confusas (Greer & McCalla, 1991a).

Na construção deste modelo, três fases podem ser percebidas: *fase de modelagem*, que envolve o entendimento do comportamento do estudante no contexto do ambiente no qual ocorre a aprendizagem, *fase de intervenção*, que engloba assistência para preencher falhas no conhecimento do estudante assim como correção de concepções errôneas, e *fase de avaliação*, que serve para validar o modelo original do conhecimento do estudante, bem como fazer os ajustes necessários (Greer & McCalla, 1991b).

Usualmente o conhecimento do estudante é modelado em comparação com o modelo do especialista. De acordo com esta relação podem-se distinguir vários tipos de modelos:

1) Modelo de Sobreposição (Overlay) - a hipótese deste modelo é que todas as diferenças entre os modelos do estudante e especialista são devidas à falta de conhecimento do estudante. O modelo do estudante é visto então como um subconjunto do modelo do especialista. Uma falha deste modelo é que não trata do conhecimento do estudante que difere do modelo do especialista. Por exemplo, se o estudante tem uma estratégia alternativa que funciona bem para um determinado problema, o tutor provavelmente tentará corrigi-la.

2) Modelo de Perturbação (Buggy) - este modelo não considera o conhecimento do estudante como um subconjunto do modelo do especialista, mas um subconjunto de uma união do domínio e outro conjunto com todas as concepções errôneas que o estudante poderia ter. Neste modelo é construída uma biblioteca de erros (bugs), e com o progresso do estudante, o modelo pode ser atualizado com vistas à presença ou ausência de erros conhecidos nesta biblioteca.

3) Modelo de Incerteza (Fuzzy) - este modelo emprega os conceitos de imprecisão e incerteza. Cada variável de conhecimento é associada com uma distribuição que possibilita ser atualizada em diferentes graus, dependendo do tipo e peso de evidência que aparece num determinado problema dado para o estudante resolver. Esta distribuição de possibilidades da variável de conhecimento é feita em vários níveis, abrangendo desde o nível de “não conhecimento” até o “pleno conhecimento”.

4) Modelo de Simulação - este modelo explica o comportamento do estudante em termos de um modelo geral e um procedimento de inferência (como o estudante interpreta observações e o modelo geral) (Greer & McCalla, 1991a).

Alguns autores diferenciam o processo de *modelagem do estudante* do processo de *modelagem do usuário*. Enquanto que o primeiro trata mais do que incluir no modelo e como diagnosticar concepções errôneas, o segundo trata mais de como adquirir modelos baseados em incerteza e como adquirir informações práticas (Paiva, 1995).

Existem ainda vários obstáculos com relação à modelagem do estudante que resultam do problema de inferir conhecimento do comportamento do mesmo. Alguns destes obstáculos são:

- O ambiente contém uma grande quantidade de incerteza e ruído;

- A inferência do estudante pode ser baseada sobre conhecimento inconsistente.

A criatividade do estudante pode gerar novos comportamentos que requerem mais sofisticação para serem interpretados (Greer & MacCalla, 1991a).

2.4.3.3 O Modelo Pedagógico

Um professor tem diversas maneiras de expor um assunto, tornando-o compreensível e interessante. A comunicação de um corpo de conhecimento para uma pessoa não segue simplesmente um protocolo de transferência de informação como acontece, por exemplo, entre dois computadores. Ela é guiada por estratégias e técnicas que são selecionadas e combinadas dinamicamente em reação às atitudes e necessidades dos estudantes.

Portanto, a tarefa do modelo pedagógico é bastante complexa. Este modelo contém o conhecimento necessário para tomar decisões sobre quais táticas de ensino deve empregar dentre aquelas disponíveis no sistema. As decisões e ações deste modelo são altamente dependentes dos resultados do processo de diagnóstico. O modelo pedagógico diagnostica as necessidades de aprendizagem do estudante com base nas informações do modelo do estudante e na solução do professor contida no modelo do especialista. Em geral, as decisões são sobre qual informação apresentar ao estudante, quando e como apresentá-la.

As representações explícitas do conhecimento pedagógico proporcionam o potencial necessário para que o sistema consiga adaptar e melhorar suas estratégias e, também, para seus componentes poderem ser reutilizados em outros domínios.

A adaptação da instrução implica uma escolha didática, que pode ser global ou local. Escolhas no nível global determinam a seqüência dos episódios educacionais e, por exemplo, decisões sobre o número de repetições a serem utilizadas. Escolhas no nível local envolvem decisões sobre quando uma intervenção é necessária, se o estudante deve ou não ser interrompido em sua atividade, e o que pode e deve ser feito ou apresentado em algum determinado momento. Isto inclui orientação no desempenho das atividades, explicações dos

fenômenos e processos, e decisões sobre quais informações serão oferecidas para reparar as deficiências dos estudantes (Wenger, 1987).

Existem diversas abordagens pedagógicas empregadas em STIs, mas a maioria dos sistemas atuais tende a implementar somente uma estratégia pedagógica. Por isso, estes sistemas não possuem um rico repertório de estratégias de ensino a serem selecionadas. Esta deficiência existe, em parte, porque a maioria das pesquisas concentram-se nos problemas de representação de conhecimento e diagnósticos, ao invés dos processos pedagógicos envolvidos no ato de ensinar.

Outra consideração é que, em geral, as estratégias pedagógicas são dependentes do contexto geral do ambiente de aprendizagem embutido no sistema. A escolha deste ambiente de ensino é ditado pela natureza do conteúdo a ser ensinado, pelo conhecimento e experiência do estudante, e pelas hipóteses sobre aprendizagem inerentes à teoria básica sobre a qual o sistema está baseado.

As decisões pedagógicas são tomadas no contexto de um ambiente educacional que determina o grau de controle sobre a atividade e sobre a interação possuídos respectivamente pelo sistema tutorial e pelo estudante (Wenger, 1987). Todas estas decisões são sutis. A ordem e maneira na qual os tópicos são tratados podem produzir experiências de aprendizagem diferentes. Em uma orientação tutorial, algumas vezes é mais eficaz deixar o estudante pesquisar a vontade ao invés de interrompê-lo, enquanto outras vezes, esta liberdade pode deixá-lo desorientado. Portanto, um processo de aprendizagem depende de uma grande variedade de fatores e o sistema tutorial deve cuidar para não destruir a motivação pessoal do estudante ou o seu senso de descobrimento. Este processo pedagógico requer grande versatilidade.

O módulo pedagógico, também chamado de modelo tutor ou instrucional, usa as informações do modelo do estudante para determinar quais aspectos do conhecimento do domínio poderiam ser apresentados para o aprendiz. Estas informações poderiam conter novos materiais, uma revisão de tópicos anteriores ou do atual. Halff (1988) define instrução, como a

forma de apresentação do conteúdo a ser ensinado, e currículo, como a seleção e o seqüenciamento do conteúdo a ser apresentado ao aprendiz. Ainda segundo Halff (1988), um STI, mesmo utilizando diferentes técnicas pedagógicas, deve apresentar as seguintes características: 1) Ter controle sobre o currículo, 2) Saber responder às questões do aprendiz, quanto ao conteúdo, 3) Ser capaz de determinar as necessidades do aprendiz.

Wenger (1987), identifica neste módulo, dois tipos de suporte:

1) **Diagnóstico** - são feitas inferências sobre o estado do aprendiz em três níveis: *Comportamental* (ignora o conhecimento do aprendiz, e foca apenas o comportamento observável), *Epistêmico* (trata com o estado de conhecimento do aprendiz, e tenta inferir aquele estado, baseado no comportamento observado) e *Individual* (são as motivações e conceitos que o estudante tem sobre o domínio).

2) **Didático** - trata da entrega do material instrucional para o estudante. É organizado em torno de quatro princípios:

Planos de ação - é um mini-curriculum usado para guiar o estudante, e prover as operações de diagnóstico.

Contextos pedagógicos - aqui os planos de ação são implementados. Podem ser: oportunistas (alguma pergunta que o estudante faz, por exemplo), ou planejados (dinamicamente controlado pelo sistema, o qual organiza a atividade e a interação).

Base de decisão - quando e como o sistema deve intervir em um objetivo didático implica às vezes na resolução de conflitos.

Nível do modelo do estudante (comportamental, epistêmico ou individual), no qual uma operação busca modificações imediatas.

Entre as estratégias de ensino mais usadas pelos STIs estão:

Método Socrático - o tutor aprofunda o conteúdo do domínio a partir de um tópico no qual o aprendiz já tenha algum conhecimento, e conduz o estudante a sucessivas questões, para formular princípios gerais sobre casos individuais. Faz uso de diálogos entre o tutor e o estudante.

Método de treinamento (coaching) - o sistema monitora as atividades do aprendiz e interrompe para dar conselhos. Algumas vezes o aprendiz lidera o processo de resolução e o tutor indica apenas sua concordância ou não. Em outras vezes o tutor orienta o aprendiz nas partes mais difíceis.

Método baseado em casos - este modelo é usado para apoiar o ensino em domínios de conhecimento vasto. Neste caso, é preciso uma grande quantidade de regras para descrever toda a área, e uma alternativa é apresentar o domínio na forma de casos-exemplo. Raciocínio baseado em casos é uma técnica de Inteligência Artificial que procura em uma base de casos, um caso mais próximo (análogo), para aplicar a solução.

Método de simulação - os ambientes que utilizam simulação, se diferenciam dos demais na natureza do modelo do domínio, que é modelado em termos quantitativos para executar cálculos das variáveis de simulação envolvidas. É um método bastante usado em sistemas de treinamento profissional, tais como: simuladores de voo, simuladores de tratamento de pacientes, etc. Halff,(1988).

2.4.3.4 O Modelo da Interface com o estudante

Na engenharia de *software*, a interface do usuário tem sido a primeira preocupação dos projetistas quando estão discutindo a criação de uma nova aplicação, pois, como afirmam Hix e Hartson, “Para os usuários, a interface é o próprio sistema”. (Hix, 1993).

Muitos princípios baseados nas teorias cognitivas têm sido propostos para projetos de interface, como resultado de pesquisas na área da interação homem-máquina. Entretanto, a meta da maioria destas pesquisas é, de um modo ou de outro, proporcionar ao usuário um alto grau de eficiência e eficácia com relação à utilização do sistema. O usuário não deve ter que se adaptar à interface do sistema, ao contrário, a interface deve ser projetada para que seja intuitiva e natural para ele aprender a utilizá-la.

Com relação aos sistemas tutoriais, enquanto o modelo pedagógico decide o momento e o conteúdo das ações didáticas, o modelo de interface cuida de sua forma final. É responsável pelo fluxo de comunicações de entrada e saída, proporcionando a comunicação em ambas direções e realizando a tradução entre a representação interna do sistema e a linguagem de interface de maneira compreensível ao estudante. Mas, a interface não tem somente uma importância para a entrada e saída da informação, ela também complementa

dados importantes sobre o processo da aprendizagem. Dependendo do domínio, estes dados podem ser utilizados para monitorar este processo.

Uma vez decidida a forma na qual o sistema apresentará um determinado tópico, a interface tem o poder de determinar o grau de compreensão desta apresentação junto aos estudantes.

Por causa do grande progresso da pesquisa em tecnologias gráficas, novas ferramentas para a construção e prototipação de interfaces podem direcionar todo o projeto do sistema.

Um outro aspecto relacionado com interfaces, e tradicionalmente relacionado com as pesquisas em IA, é o processamento da linguagem natural. Este ramo de pesquisa ainda é um grande obstáculo a ser vencido e, em seu estágio atual, limita as possibilidades de uma comunicação totalmente amigável com os computadores.

É importante salientar que em uma interação com o STI, o estudante não irá somente aprender o conteúdo das lições, mas também terá que aprender como utilizar o sistema, portanto, a facilidade de uso deve ser uma das considerações principais no projeto destas interfaces. Uma interface consistente ajudará a reduzir a carga cognitiva sobre o estudante (Shneiderman, 1992).

O modelo de interface com o estudante é responsável por processar o fluxo de informações entre o sistema e o usuário ou aprendiz. Traduz a representação interna do sistema numa linguagem compreensível para o aprendiz. Embora a interface opere em cooperação com os outros módulos, suas decisões requerem um tipo diferente de conhecimento (Wenger, 1987).

2.4.4 “Arcabouços” e Sistemas de Autoria

Um “arcabouço” para STIs é um ambiente geral para se construir STIs, enquanto que um sistema de autoria para STIs é um “arcabouço” para STIs juntamente com uma interface do usuário que permite não programadores formalizar e visualizar seu conhecimento (Murray, 1999).

Murray divide em várias categorias os STIs que utilizam sistemas de autoria:

1) *Seqüenciamento de curriculum e planejamento*: sistemas de autoria nesta categoria organizam as unidades de instrução numa hierarquia de cursos, lições, apresentações, etc., que são relacionadas por pré-requisitos.

2) *Estratégias de ensino*: estes sistemas tendem a serem similares aos descritos acima, entretanto codificam estratégias de fina granularidade e têm a habilidade para representar múltiplas estratégias de ensino, selecionando qual a mais apropriada para uma dada situação.

3) *Simulação e treinamento*: a diferença destes sistemas em relação aos descritos acima, é que o estudante é colocado numa situação de “aprender fazendo”.

4) *Sistemas especialistas e tutores cognitivos*: estes sistemas incluem modelos cognitivos baseados em regras do domínio do especialista. Aqui os estudantes trabalham na resolução de problemas e sub-problemas dentro de um objetivo, e recebem um retorno quando seus comportamentos divergem do modelo do especialista.

5) *Hipermídia inteligente*: em muitos dos sistemas baseados na Web hoje, o nível de interatividade e fidelidade disponível para o estudante ainda é baixo. Ao contrário das outras categorias, estes sistemas precisam gerenciar os “hiperlinks” entre unidades de conteúdo diversas, bem como a forma e a seqüência do próprio conteúdo.

O principal objetivo então, em se usar sistemas de autoria nos STIs, é que torna o processo de construção mais fácil e reduz os custos. Uma outra questão importante que vem sendo tratada por pesquisadores é o reuso. Isto pode ser feito em vários níveis (Murray, 1999):

- *Base de conhecimento* (utilizando partes de um domínio particular);
- *Bibliotecas de ensino* (exercícios e testes desenvolvidos no mesmo domínio);

- *Arquiteturas* (arquiteturas genéricas para uma vasta gama de domínios).

Os sistemas de autoria (Murray,1999) podem ser colocados em duas categorias: *sistemas de orientação pedagógica*, e *sistemas orientados à performance*. O primeiro tipo foca a atenção em como seqüenciar e ensinar um conteúdo pré-determinado, e o segundo tipo foca a atenção num ambiente no qual os estudantes possam aprender habilidades, praticando eles mesmos, por meio de interações com o sistema.

2.4.5 Evolução para os Ambientes Interativos de Aprendizagem

As críticas de vários autores em relação aos STIs dos anos 70 e 80, sugerem que estes sistemas davam demasiado controle ao tutor, assim como o fato de que eles eram modelados numa abordagem instrucionista, enfatizando a transferência para o estudante, do conhecimento que o tutor possuía (Kearsley, 1987).

McLoughlin e Oliver (1995) definem Ambiente Interativo de Aprendizagem como um ambiente em que o controle do processo de aprendizagem é feito pelo aprendiz, incluindo aí a escolha da seqüência e quantidade da informação a ser processada.

O conceito de interação é vista, segundo Giardina (1992), como um complexo conceito que envolve controle, adaptação e inteligência. Interatividade não é somente definida em função dos aspectos técnicos do ambiente, mas também em relação às ações e decisões do aprendiz e ajustes às diferenças individuais.

A questão central passa a ser então: como manter o controle por parte do aprendiz (usuário) e ao mesmo tempo ter um sistema que oriente implicitamente a sua navegação pelo conteúdo de um hipertexto. Entre as alternativas propostas em Kommers et al, (1996) estão:

- Prover o aprendiz de um mecanismo de orientação que mostre o que é central e periférico no nível de informação.
- Especificar a função do hipertexto criando uma série de lições.

- Restringir o número de estratégias de aprendizagem fazendo com que o aprendiz escolha o estilo mais adequado.

2.4.6 Classificação

Segundo Wilson (1996), os Ambientes de Aprendizagem são opostos aos ambientes instrucionais, no sentido de promover uma idéia mais flexível de aprendizagem na qual o aprendiz pode desenvolver habilidades mais relevantes para a resolução de problemas.

Murphy (1997) propõe algumas abordagens para os Ambientes de Aprendizagem:

Ambientes de Aprendizagem Baseados em Computador - ambientes nos quais as atividades propiciadas pelo computador são integradas a outras atividades, havendo uma mútua influência;

Ambientes de Aprendizagem com Multimídia Interativa - ambientes não estáticos, com papéis não fixados para o professor e o aprendiz, com equilíbrio dinâmico, onde a natureza e a mudança da informação dependem da situação, do contexto do aprendiz e das necessidades individuais;

Ambientes de Aprendizagem Construtivista - se baseiam nos seguintes princípios: prover experiência no processo de construção do conhecimento, prover colaboração e múltiplos modos de representação, e encorajar atividades reflexivas;

Ambientes de Aprendizagem Construcionista - se baseiam no conceito de se produzir o máximo de aprendizagem com o mínimo de ensino. Representar frações por meio de jogos, por exemplo, ilustra a idéia de construção mental (construções no mundo como suporte para as da mente).

Wilson (1996) classifica em três categorias os ambientes de aprendizagem:

Micromundos, Ambientes de aprendizagem baseados em sala de aula e Ambiente virtual.

2.4.7 Multimídia, Hipertexto e Hipermissão

Multimídia é a integração de várias mídias tais como texto, gráficos, animação, som e vídeo. Schwier e Misanchuk (1993) distinguem três níveis de interação para instrução multimídia: *reativa* (resposta a um dado estímulo), *pró-ativa* (geração de construções únicas), e *mutual* (uso de Inteligência Artificial). Neste último nível o sistema se adapta ao progresso do aprendiz, além de construir e refinar o ambiente para este.

Hipertexto é um método não linear e não sequencial de mostrar texto, gráficos, animação, som e vídeo e que tem as seguintes características: nós de informação, ligações entre os nós, estrutura organizada como uma rede de idéias, controle dinâmico por parte do usuário e acesso de múltiplos usuários. *Hipermissão* é a união de hipertexto e multimídia (Davies, 1997).

Segundo Padgett (1997), entre as vantagens em se usar hipermissão estão: o favorecimento da memória associativa, a apresentação de uma grande variedade de perspectivas nos tópicos e controle por parte do aprendiz. Alerta, contudo os *designers* de hipermissão para que não haja uma sobrecarga de informação e desvios pelo excesso de *links*.

A estrutura básica de um sistema multimídia é composta de nós e ligações (*links*). Esta forma não linear de dispor a informação é chamada de modular. Um nó pode ser construído de duas maneiras: pelo projetista ou pelo usuário. O primeiro caso é mais apropriado para sistemas nos quais o objetivo e conteúdo não devam ser mudados pelo usuário. O segundo caso é usado quando há entrada de dados pelo usuário, como respostas a questões, pedidos de informações ou comentários (Kommers et al, 1996).

Já os *links* podem ter a seguinte classificação:

- *Link Contextual* - junta as várias partes do hipertexto e permite o usuário achar a informação que precisa. O *link* contextual se subdivide em dois tipos:
 - *Sequencial* - cria um caminho pré-determinado onde o usuário tem apenas duas opções: próximo nó ou nó anterior;

- *Relacional* - permite ao usuário buscar a informação ligada por elementos comuns, embora não de uma forma seqüencial. Existem três tipos: *associativo* (procura pela informação relativa a uma palavra, frase ou nó específico), *elaborativo* (tipo de link seqüencial que provê informação mais detalhada sobre um tópico específico), e *hierárquico* (liga a informação de uma maneira progressiva e estruturada).
- *Link de Suporte* - são *meta-links* que provêm ajuda para o usuário acessar e aprender do hipertexto Kommers et al, (1996).

2.4.8 Aspectos que devem ser considerados em Sistemas Tutores Inteligentes

Sistemas Tutores Inteligentes que incorporam técnicas variadas de IA podem ser considerados tecnicamente bem projetados e implementados. Entretanto, nos últimos tempos, uma preocupação que tem sido considerada constante está relacionada à adequação pedagógica dos mesmos. É importante ressaltar que este fator estende-se a qualquer *software* com propósitos educacionais e não apenas aos STI's.

É interessante considerar que para que um *software* educacional seja efetivo e realize os seus objetivos pedagógicos é necessário que o seu desenvolvimento reúna especialistas das áreas de Educação e Informática. Desta forma, para que possa existir este trabalho interdisciplinar, os profissionais que se dedicam ao desenvolvimento desse tipo de *software*, devem buscar alguns referenciais teóricos que fundamentem o seu trabalho. Esses referenciais teóricos têm a sua origem nas áreas de Educação e de Psicologia e estão relacionados às teorias de aprendizagem. A importância dessas teorias é que os procedimentos e princípios de ensino que estão embutidos no conjunto de táticas e estratégias de ensino de um *software* educacional estão diretamente relacionados a elas.

Não é objetivo deste trabalho explorar as teorias psicológicas e pedagógicas a partir de seus autores e sim compreender algumas tendências relacionadas ao desenvolvimento de STI's. De modo geral, as teorias de ensino-aprendizagem envolvidas no projeto de STI's, estão relacionadas a determinadas abordagens pedagógicas. Algumas das abordagens que podem ser encontradas em Giraffa, (1995) e consideradas neste trabalho são as seguintes: comportamentalista (*behaviorista*), baseada principalmente nos trabalhos de *Burrhus Frederic Skinner*; humanista, baseada nos trabalhos de *Carls Ranson Rogers*; cognitivista, baseada nos trabalhos de *Jean Piaget* e *Jerome Seymour Brunner* e sócio-cultural, baseada principalmente nos trabalhos de *Paulo Freire* (no âmbito do Brasil).

Assim, se o *software* educacional desenvolvido segue uma das diferentes abordagens pedagógicas, qualquer que seja ela, é necessário que se tenha o conhecimento necessário desta, a fim de compreender melhor a abrangência, as limitações e as aplicações que o *software* pode atender. Maiores detalhes em Giraffa (1995).

2.4.9 - Estado da Arte: Exemplos de Sistemas Tutoriais Inteligentes

As pesquisas em sistemas inteligentes têm produzido e certamente continuarão proporcionando um campo de visão para os problemas relacionados com aprendizagem e instrução.

O desenvolvimento de um *software* educacional inteligente atualmente requer uma grande quantidade de esforço concentrado e uma diversidade de conhecimentos e técnicas. Além disso, muitos desses programas têm sido indevidamente rotulados “inteligentes”. Portanto, um estudo comparativo de STIs é importante para os profissionais das diversas áreas de conhecimento envolvidas na construção destes sistemas. Certamente, novas pesquisas trarão avanços consideráveis em áreas como interação homem-máquina e aprendizagem humana e de máquina.

A seguir serão descritos de forma comparativa através do quadro 1 alguns dos esforços historicamente importantes no desenvolvimento de Sistemas Tutoriais Inteligentes. A maioria destes STIs tem sido extensivamente documentados, como por exemplo, em Wenger, (1987).

Quadro 2 Simplificação dos Sistemas

Sistema	Referência	Conhecimento do Domínio	Conhecimento especialista	Particularidade
SCHOLAR	Carbonell 1970	Rede Semântica	Objetos e conceitos geográficos sobre a América do Sul	O sistema pode determinar a relação semântica entre 2 nodos
SOPHIE	Brown 1975	Modelo de Simulação	Conceitos de equipamentos eletrônicos com defeito	Permite que o estudante tenha uma relação um-para-um com o especialista baseado em computador
WEST	Burton 1979	Jogo educacional para computadores	Técnicas de Aritmética	West contém não somente uma representação do conhecimento do especialista, mas também técnicas tutoriais para utilizar a informação produtivamente.
GUIDON	Clancey 1987a	Método de caso: um dialogo de iniciativa mista	Diagnósticos de doenças Infecciosas	Um dos primeiros na tentativa de transformar um sistema especialista em Tutor inteligente.
HYDRIVE	Kaplan 1995.	Sistema Tutor Inteligente	Problemas de sistemas hidráulicos de um avião F15	Incorpora Multimídia. O sistema, quando necessário, pode selecionar um conteúdo de vídeo, a partir de um disco laser, e apresentá-lo.
STCEQ	Reis 2001	Sistema Tutor Inteligente	O Ensino de Controle Estatístico da Qualidade	Apresenta oito módulos que se relacionam de forma diferente de acordo com o tipo de interação escolhido pelo usuário. Contem 82 tutoriais e 36 problemas implementados com um modulo simulador que gera os resultados para os problemas.
MCOE	Giraffa 1999	Jogo educacional multimídia	Ecossistema	um sistema de reprodução em equilíbrio, até que a intervenção de poluentes provoquem alterações no seu estado normal. Estes poluentes aparecem de forma aleatória ao longo do jogo e são combatidos através de ferramentas do personagem escolhido por cada aprendiz.

2.4.9.1 A arquitetura *MutAntIS*

Apresentaremos agora a arquitetura *MutAntIS* e uma comparação com a arquitetura GIA, para que possamos ter conhecimento de outras arquiteturas utilizadas na construção dos sistemas tutores e outras formas de estruturar um sistema tutor inteligente diferente da estrutura dos 4 modelos estuda anteriormente

A arquitetura *MutAntIS*, proposta por Azevedo, (1999), especifica uma arquitetura multi-agentes para a construção de sistemas tutores inteligentes, descrevendo as características específicas de cada um dos agentes que fazem parte da arquitetura e o esquema de comunicação entre eles. A idéia inicial desta arquitetura veio da arquitetura GIA (Cheikes, 1995). Tal arquitetura representa uma proposta para utilização de agentes inteligentes em sistemas tutores inteligentes. Azevedo (1999) ressalta que a diferença entre a arquitetura *MutAntIS* e a arquitetura GIA é que a GIA apresenta de uma forma geral a arquitetura multi-agente, isto é, ela descreve as características gerais dos agentes e o esquema de comunicação entre eles. Entretanto, ela não entra em detalhes sobre quais os agentes devem fazer parte da arquitetura, e as características específicas de cada um desses agentes. A arquitetura *MutAntIS*, seguindo o modelo da arquitetura GIA (com os tipos de agentes propostos e o esquema de comunicação), especifica quantos e quais os agentes devem fazer parte da arquitetura, além das características específicas de cada um. A arquitetura *MutAntIS* é composta por seis agentes inteligentes, representados na Figura 10.

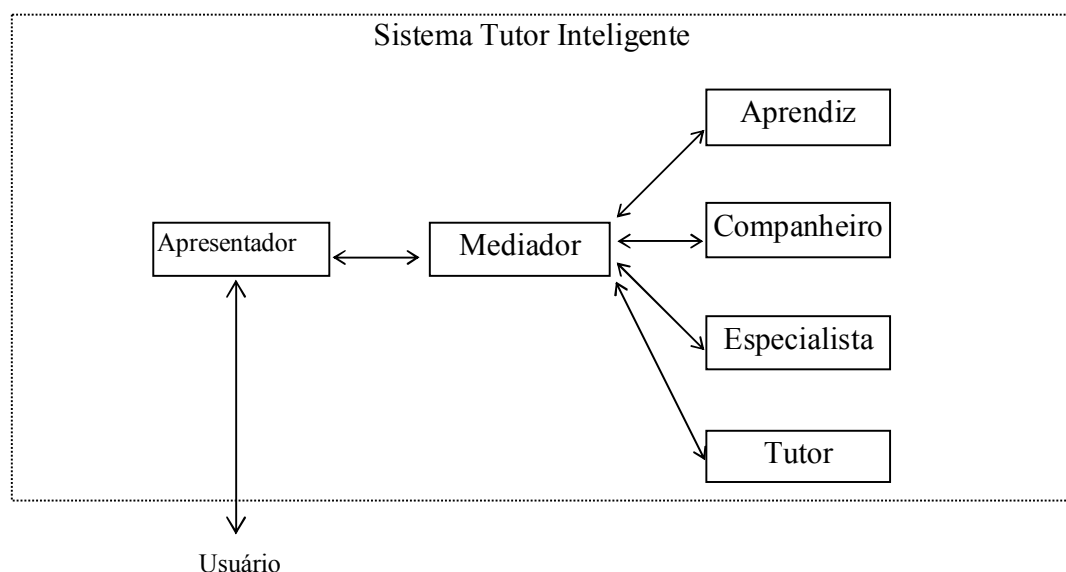


Figura 10 – Arquitetura *MutantIS*

A seguir, uma breve descrição de cada um dos agentes inteligentes da arquitetura *MutAntIS*:

Agente Apresentador: tem a função de realizar a interface do STI com o usuário, oferecendo todos os serviços relativos à apresentação de informações ao usuário e todos os serviços relativos à obtenção de informações fornecidas pelo mesmo.

Agente Mediador: a função deste agente é atuar como um elo de ligação entre todos os agentes, fazendo um roteamento de mensagens entre eles. Quando o STI for inicializado, todos os agentes devem enviar ao *Mediador* os serviços que eles oferecem (as tarefas que eles podem executar) e as necessidades de cada um (as informações de que precisam). Com isso, o *Mediador* será capaz de “saber” quais os serviços oferecidos por cada agente e quais as necessidades de cada um.

Agente Aprendiz: este agente é responsável por criar e manter modelos dos aprendizes (usuários) que utilizam o STI. O *Agente Aprendiz* deve também fornecer informações ao *Agente Tutor*, para identificação de deficiências no modelo do aprendiz e para controle do progresso do usuário. O *Agente Aprendiz* pode também fornecer informações para ajudar a

geração de problemas e a adaptação de explicações. O modelo do aprendiz é construído dinamicamente, à medida que o usuário utiliza o STI.

Agente Companheiro: o papel deste agente é atuar como um companheiro de estudo do usuário, podendo atuar de diversas formas (fornecendo dicas para auxiliar o usuário na resolução de problemas, na realização de tarefas, nas questões elaboradas pelo STI, por exemplo).

Agente Especialista: contém uma representação do conhecimento a ser comunicado ao usuário, agindo como a fonte do conhecimento a ser apresentado. Ele deve ser capaz de gerar explicações, respostas, tarefas e questões que devem ser apresentadas ao usuário.

Agente Tutor: projeta e regula as interações instrucionais com o usuário. É ele quem decide as atividades pedagógicas que serão utilizadas: avisos, suporte, explicações, tarefas diferentes, entre outras. O *Agente Tutor* determina o que o STI vai apresentar ao usuário, como isso deve ser realizado e quando deve ser feito. Para isso, ele age de acordo com informações obtidas do *Agente Aprendiz* e do *Agente Especialista*.

2.5 Conclusões

O estudo da probabilidade nos mostra sua ampla utilização no dia a dia das pessoas, pois mesmo sem notar as pessoas utilizam a probabilidade para tomar decisões do cotidiano, ao escolher um caminho levando em consideração a probabilidade dele estar ou não engarrafado, ou quando se decide por levar ou não um guarda-chuva ao sair de casa. Esses e outros motivos fazem da probabilidade um conceito muito utilizado e estudado por vários e diferentes grupos.

Os sistemas tutores inteligentes proporcionam uma estrutura alternativa no aprendizado de conceitos buscando modelar aspectos da tutoria humana para contribuir no processo de aquisição de conhecimento e aprendizagem fornecendo maior controle por parte do usuário.

Os sistemas tutores inteligentes surgiram há algum tempo e vem incorporando novos recursos com a evolução das ferramentas buscando o aperfeiçoamento dos sistemas tutores.

Entre os sistemas tutores pesquisados aparentemente não foi encontrado um sistema que tivesse as características implementadas neste trabalho, um sistema tutor inteligente para auxiliar no aprendizado de conceitos de probabilidade e estatística.

No próximo capítulo será apresentada a metodologia empregada para o desenvolvimento de um sistema tutorial inteligente para auxiliar no ensino de Probabilidade.

3 METODOLOGIA

Neste capítulo será apresentada a metodologia desenvolvida para este trabalho, suas principais características e as decisões tomadas durante a construção do sistema. O capítulo traz também uma abordagem generalista sobre a linguagem de programação utilizada para o desenvolvimento do sistema.

3.1 Pesquisa de Sistemas Tutores Inteligentes

A pesquisa foi desenvolvida nas seguintes etapas: organização do material a ser pesquisado, leitura, delimitação de conteúdo e sua interpretação.

A organização do material a ser pesquisado inclui a observação e transcrição de notas relevantes para a pesquisa.

A interpretação dos dados segundo Marconi e Lakatos (1996, p.32) “é a atividade intelectual que procura dar significado mais amplo às respostas, vinculando-se a outros conhecimentos”.

3.2 Metodologia desenvolvida

No início da pesquisa a intenção era a produção de uma metodologia de ensino para o aprendizado da Estatística e da Probabilidade implementados em um *software* educacional. Buscou-se então estudar as ferramentas de educação que poderiam ser implementadas para o ensino da Estatística e Probabilidade, e o sistema tutor inteligente apresentava as características da tutoria humana desejadas em um *software* educacional, capaz de auxiliar o

aprendiz no processo de construção do conhecimento, possibilitando ao aprendiz um controle maior neste processo. A escolha direcionou a pesquisa para o estudo dos sistemas tutores inteligentes e sua forma de construção. Observou-se que não existe um parâmetro definido para a construção de um sistema tutor inteligente utilizado na educação: a análise dos sistemas existentes proporcionou uma avaliação dos mesmos e algumas características que poderiam ser implementadas no sistema tutor inteligente proposto neste trabalho. No entanto, aparentemente nenhum dos sistemas tutores inteligentes estudados, apresentava as características desejadas inicialmente.

O próximo passo foi a compilação dos conceitos de probabilidade e estatística e a estruturação do sistema. Percebendo que a probabilidade e a estatística são compostas de conceitos amplos e complexos, surge a necessidade da escolha dos conceitos que seriam apresentados no sistema, mas, buscando estruturar o sistema de maneira a permitir a inclusão de novos conceitos quando desejado. O conteúdo deveria ser então introdutório o que levou a escolha de três conceitos básicos: Introdução à Probabilidade, Modelos para Variáveis Aleatórias Discretas e Modelos para Variáveis Aleatórias Contínuas.

O conteúdo para inclusão no sistema foi retirado do livro Estatística: para curso de Engenharia e Informática. (BARBETTA, P. A. , REIS, M. M., e BORNIA, A. C., 2004). Uma didática de apresentação já elaborada e testada em sala de aula, para agilizar a construção do sistema.

Como a estrutura didática contida no livro incluía exercícios ao final de cada capítulo buscou-se implementar esses exercícios no sistema. A idéia inicial era que o sistema tivesse uma estrutura de problemas e o aprendiz pudesse entrar com as informações para a construção do problema e o sistema ajudaria o aprendiz a resolvê-lo. Devido à complexidade de desenvolvimento optou-se por manter a estrutura de exercícios fixos no sistema, mas com a possibilidade de que o sistema tutor inteligente corrija o exercício quando solicitado pelo aprendiz.

Como o objetivo do sistema é ser um auxiliar do professor no ensino-aprendizado de probabilidade e estatística, buscou-se implementar uma forma do sistema trocar informações com o professor. Isso foi possível com a elaboração de um relatório onde o tutor apresenta ao professor todos os conteúdos já acessados pelo aprendiz no sistema, para que o professor possa acompanhar o aprendiz em seu aprendizado. Este relatório é montado pelo sistema tutor através do monitoramento das ações do aprendiz dentro do sistema. Para que isso fosse possível houve a necessidade de acesso ao sistema através de um login, de modo que o sistema pudesse reconhecer qual aprendiz estava utilizando o tutor, e então arquivar seus passos dentro do sistema: a cada ação do aprendiz grava-se em uma base de dados uma linha que indica o que ele fez. Quando o relatório é solicitado, o sistema faz a leitura desta tabela e apresenta-a ao professor em forma de relatório. Isso será melhor explicado no próximo capítulo.

Para a construção do sistema optou-se pela a linguagem Delphi, por se tratar de uma linguagem de programação de fácil utilização, sendo orientada a objeto o que permite a reutilização de código. Como o sistema tem como uma das propostas a implementação de novos conceitos, o Borland Delphi® permite manter a estrutura e reutilizar código na construção dos novos conceitos, sem que se faça necessário a elaboração de um novo algoritmo. O banco de dados utilizado foi o Microsoft Access®, por se tratar de um sistema onde a base de dados é relativamente pequena e simples. Esta ferramenta demonstrou ser capaz de suprir as necessidades do sistema, com a vantagem de não ser necessária sua instalação, por se tratar de um aplicativo já incorporado ao Windows, facilitando a implementação e configuração do sistema tutor inteligente desenvolvido. Maiores detalhes serão vistos nas próximas seções.

3.3 Tecnologia Utilizada para seu desenvolvimento

Conforme já mencionado, para o desenvolvimento do sistema utilizou-se como linguagem de programação o ambiente Borland Delphi®. Devido às próprias características e natureza do sistema foi utilizado o gerenciador Microsoft Access® como banco de dados, para

facilitar a instalação do sistema. O gerenciador se comunica com o sistema via ODBC (Open DataBase Connectivity).

3.3.1 Borland Delphi®

O Borland Delphi® é um ambiente de desenvolvimento de aplicações, orientado a objeto, que permite o desenvolvimento de poderosas aplicações baseadas no Microsoft Windows® com o mínimo de codificação. Ele também oferece ferramentas de desenvolvimento, tais como templates de aplicações e forms, que lhe permitem criar e testar rapidamente o protótipo de suas aplicações. É possível utilizar o conjunto de componentes e código gerado para transformar seus protótipos em aplicações robustas que satisfaçam suas necessidades.

O Borland Delphi® também oferece ferramentas de bancos de dados que permitem desenvolver aplicações cliente/servidor e relatórios. As ferramentas de bancos de dados permitem a visualização dos dados dinamicamente durante o desenvolvimento, para que se verifique imediatamente se os resultados das *queries* estão de acordo com as necessidades. Para se ter uma idéia da orientação a objeto no ambiente e como isso torna possível a reutilização de código; serão abordados neste item conceitos pertinentes à orientação a objetos.

3.3.1.1 Orientação a Objeto em Borland Delphi®

A ferramenta em questão é um ambiente de desenvolvimento de aplicações, orientado a objeto (POO), que permite o desenvolvimento de aplicações baseadas no Windows com um mínimo de codificação.

O ambiente possui sua própria implementação de Orientação a Objetos. Algumas vezes é mais flexível, outras vezes mais restrita. A seguir, será mostrado como implementar os conceitos de Orientação a Objetos no Borland Delphi® .

3.3.1.2 Classes

As Classes se referem a tipos de dados, definidos pelo usuário, que têm um estado (sua representação) e algumas operações (seu comportamento).

Uma classe possui:

- **Finalidade** - os objetos pertencentes a uma classe sempre possuem um objetivo comum.
- **Hereditariedade** - O Borland Delphi® possui mais de 500 classes de objeto, e toda nova classe deve ser herdeira de uma já existente. A classe base, da qual derivam todas as outras classes (direta ou indiretamente) é TObject. Herdar significa que a nova classe pode utilizar todas as características da classe antecessora além de poder ter as suas próprias, está é a relação de herança.
- **Propriedades** - Semelhante às variáveis de memória, as propriedades armazenam dados sobre o objeto que afetam o seu funcionamento ou sua aparência.
- **Eventos** - São também propriedades, mas armazenam o endereço de memória onde está uma **procedure** ou **function**. Estas são executadas automaticamente quando algum tipo de ação ocorre com o objeto.
- **Métodos** - São rotinas internas do objeto que nos prestam algum serviço, executando alguma tarefa.

The diagram illustrates the Delphi syntax for defining classes and objects. It includes three red annotations with arrows:

- Subclasse**: An arrow points to the `NomeDaClassePai` parameter in the `class` constructor.
- Classe**: An arrow points to the `class` keyword.
- Instância**: An arrow points to the `NomeDoObjeto` variable in the `var` declaration.

```

type NomeDaClasse = class(NomeDaClassePai)
    ... {definição dos atributos}
    ... {definição dos métodos}
end;

{ Instância a classe, ou seja, declara o objeto }
var NomeDoObjeto : NomeDaClasse;
  
```

Figura 7 - Sintaxe para definição de Classes e Objetos em Delphi

É possível observar a hierarquia de classes através dos próprios componentes que o Delphi possui, por ser Orientado a Objetos, como apresenta a Figura 8:

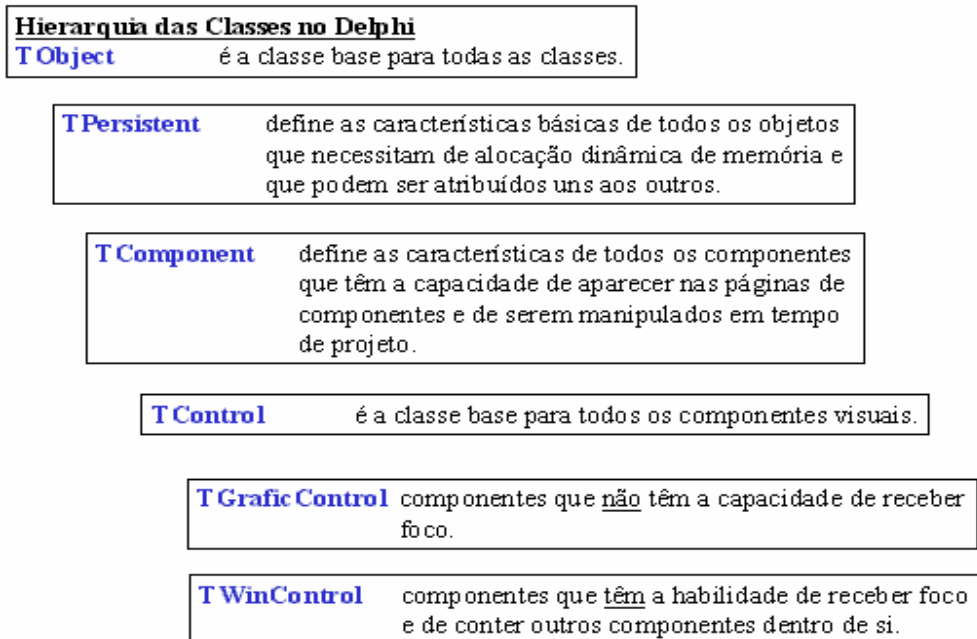


Figura 8 –Hierarquia das Classes no Delphi

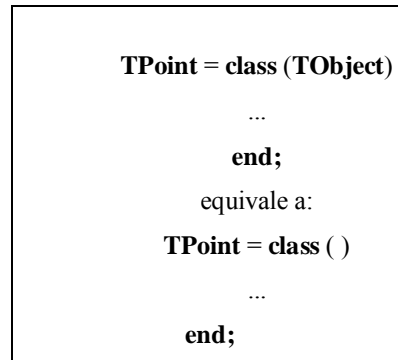
Classes são modelos a partir dos quais são gerados muitos objetos. Uma classe é na verdade a definição das características que muitos objetos irão conter. Por exemplo, pode haver uma classe Ponto, contendo os campos X, Y e Z (sua localização no espaço), mas haverá muitos objetos em um programa que tenham esta classe como referência, cada um com seu conjunto específico de X, Y e Z, como, por exemplo, o Ponto Origem (X=0, Y=0, Z=0) e o ponto A (X=13, Y=22, Z=40). Os pontos Origem e A também podem ser chamados de instâncias da classe Ponto.

Pode-se ver um exemplo no Quadro 2 de como uma classe é implementada no ambiente: a classe TPoint – Uma convenção que programadores Delphi seguem é colocar a letra T antes da definição de qualquer classe para indicar seu nome e utilizar preferencialmente o inglês em seus programas. Isto facilita a compreensão dos programas por programadores do mundo todo

Type
TPoint = Class (TObject)
Private
{Nível zero de visibilidade}
Protected
{Nível de visibilidade médio, visível a classes derivadas}
Public
{Nível máximo de visibilidade, visível a todas as classes}
Published
{Equivale a public, utilizado em tempo de design. O que houver nesta seção aparece no Object Inspector. Há regras especiais para utilização desta seção}

Quadro 3 Implementação de uma classe

A classe acima define uma classe **TPoint** derivada da classe **TObject**. Todas as classes no ambiente são derivados do **TObject**, se não for especificado nenhuma derivação, o Delphi automaticamente assumirá **TObject**, conforme apresentado no Quadro 3:



Quadro 4 Definição de classe

As seções mostradas nos quadros anteriores definem vários níveis de visibilidade. Esta é uma característica muito importante em objetos. A visibilidade permite que uma classe proteja sua implementação da interferência de outras classes, programas e qualquer outra coisa que a utilize.

Existem três níveis de visibilidade (published é um caso especial de public):

- **Private** indica que o que estiver contido nesta seção, sejam métodos ou campos, só será visível e utilizado pela classe. Somente a própria classe poderá empregar o que estiver contido nesta seção. Isso permite que a classe esconda seus métodos internos ficando livre da interferência de outras classes e programas.
- **Protected** indica um nível um pouco menor de privacidade. O que houver nesta seção somente será visível pela classe e por suas classes derivadas. Nenhuma outra classe, que não as derivadas, e nenhum programa poderá acessar esta seção. Isto é especialmente útil para implementar métodos que serão superpostos em classes posteriores.
- **Public** indica um nível de visibilidade máximo. Qualquer parte de qualquer classe ou programa pode acessar esta seção da classe. Esta seção é útil para a implementação de métodos utilitários da classe.
- **Published** é uma especialização de public. As regras de visibilidade são as mesmas. Mas nesse local só podem ser definidos dois tipos especiais de campos e métodos. Campos chamados propriedades (properties) e métodos de

evento (event handlers - properties especiais). Os campos definidos nesta seção irão aparecer na caixa Object Inspector do Delphi permitindo sua utilização em tempo de produção do programa e não somente em run-time.

3.3.1.3 Herança (*Reaproveitamento de código*)

A herança é um meio de permitir que código seja reaproveitado em uma classe. Em Delphi, a herança é extremamente simples. Basta indicar, na declaração da classe, qual será o seu "pai", ou seja, de qual classe a nova classe declarada herdar as características.

Construtores e Destrutores - Um conceito implementado por quase todas as linguagens de programação é o conceito de criação e destruição de objetos. Este conceito é implementado no ambiente com a utilização de palavras reservadas constructor e destructor. Os construtores são métodos especiais chamados antes da utilização de um objeto (quase sempre chamados automaticamente pelo Delphi) e servem para alocar memória para o objeto e inicializar suas variáveis internas. Os destrutores (quase sempre chamados automaticamente pelo Delphi) destroem o objeto na memória e antes que isso seja feito, executam ações a critério do usuário.

Os destrutores são especialmente úteis para destruir campos que foram criados no construtor.

Encapsulamento – O encapsulamento é o fato de um objeto esconder suas propriedades e ajudar nos detalhes da implementação. É possível ir um passo adiante utilizando o conceito Delphi de property. Uma property é o equivalente de um campo, visível externamente, mas escondendo seus detalhes de implementação.

3.3.1.4 Métodos virtuais e dinâmicos. Sobreposição (*override*) de métodos

Este conceito é usado na implementação de polimorfismo. O polimorfismo usa uma forma de overriding chamada de ligação dinâmica. O overriding estático permite a alteração de parâmetros porque elimina totalmente o método anterior. Entretanto, o overriding dinâmico

exige que os métodos tenham a mesma sintaxe, empregando o mesmo nome, parâmetros (tipos e ordem) para que ele seja possível. Isso pode parecer uma limitação, mas não é. O uso da mesma sintaxe permite que uma parte externa que empregue o polimorfismo não tenha necessidade (e o polimorfismo seria impossível se não fosse isso) de saber o método a chamar em cada classe para executar aquela função. Isso permite que uma classe seja usada como se fosse outra (ou seja, polimorfismo).

O polimorfismo é feito através do uso das palavras chaves **virtual**, **dynamic**, **override**, **as** e **is**.

O uso da palavra **virtual** após a definição de um método indica ao Delphi que ele não ligue os métodos na forma de endereços até que o programa comece a executar. Ou seja, quando um local no programa chama um método, se for à primeira vez, seu endereço é calculado por meio de tabelas e depois ele é chamado. Isso permite que o compilador ignore as chamadas que aparentemente são inválidas, mas que na verdade executarão a tarefa correta. A palavra **dynamic** tem o mesmo sentido, somente muda a forma com o cálculo.

Quando um método é descrito como **virtual** (ou **dynamic**), todas as suas reimplementações em classes futuras terão que ser exatamente com a mesma sintaxe. Se for mudada qualquer coisa, ordem de parâmetros, tipo ou outro item, o método terá perdido sua característica de **virtual**.

A palavra **override** é utilizada na redefinição do método em classes derivadas. Ou seja, quando o método passar por uma reimplementação deve-se acrescentar a ele a palavra **override**, de modo que o compilador identifique o fato de que este método é **virtual**.

3.3.1.5 ODBC (*Open DataBase Connectivity*)

ODBC é uma especificação projetada pela Microsoft para permitir que aplicações feitas for Windows acessem múltiplos dados de uma forma simples, desconsiderando os formatos dos arquivos de dados, simplificando assim o acesso a informações de banco de dados diferentes.

O SQL (Structured Query Language) é usado como método de acesso aos dados através do ODBC. O ODBC traz também com vantagem desvincular as aplicações de mudanças de versões na base de dados. Mudanças nos serviços da rede, servidores e base de dados não impactarão sobre aplicações usando ODBC.

A Microsoft, junto com alguns outros fabricantes de banco de dados, projetou o driver ODBC para prover um padrão, consistente e simples para conexão a bancos de dados. Fornecedores de drivers ODBC seguem as especificações de forma que qualquer aplicação ODBC possa ter acesso os dados.

Aplicações compatíveis com ODBC: Microsoft SQL Server®, Microsoft Access®, Microsoft Word®, Microsoft Excel®, Powerbuilder, Borland Delphi®, Cristal Pro Report Writer®, Visual Basic e centenas de outros.

3.3.2 Borland Delphi® sua utilização

O Delphi demonstrou ser uma ferramenta satisfatória na construção do sistema tutor inteligente, seus componentes de orientação a objeto foram utilizadas de forma a facilitar e agilizar o desenvolvimento do sistema.

A resolução dos exercícios foi feita utilizando a técnica de polimorfismo onde o procedimento para resolvê-los é o mesmo, o que muda são os parâmetros passados pelo sistema ao procedimento.

O relatório de aprendizes foi feito utilizando um componente do Delphi chamado QuickReport que facilita a construção dos relatórios buscando nas tabelas as informações que serão mostradas nos campos do relatório.

A comunicação com o banco dados via ODBC é intermediada pelo componente Database (que inclui todas as tabelas do sistema) e o componente Query que é o responsável pelo SQL (Sequencial Query Language) do banco de dados.

O código para gravar a tabela de acompanhamento nos diferentes momentos da utilização do sistema é sempre o mesmo, bastando para isso fazer a chamada do procedimento que possui este código, permitindo assim a reutilização do código.

3.4 Conclusões

Quando se desenvolve um projeto é necessário tomar decisões e fazer escolhas de acordo com os propósitos a serem alcançados.

Para desenvolver este trabalho foi necessário seguir uma metodologia baseada em pesquisas realizadas para esclarecer dúvidas e delimitar o enfoque deste trabalho. Após a decisão de desenvolver um sistema tutor inteligente para ensino da probabilidade e estatística, tornou-se necessária a escolha de uma ferramenta para seu desenvolvimento e o Delphi mostrou ser uma ferramenta eficaz para o desenvolver o sistema. Suas facilidades apresentadas anteriormente fazem do Delphi uma linguagem de fácil utilização possibilitando assim diminuir o trabalho de programação.

4 SISTEMA

Este capítulo descreve o sistema desenvolvido que é o resultado da pesquisa e da metodologia realizada neste trabalho. O sistema tutorial para auxiliar no aprendizado de conceitos de probabilidade e estatística procurando ser um auxiliar do professor no processo ensino-aprendizagem é aqui apresentado de forma a demonstrar suas funcionalidades.

4.1 Modelo do sistema desenvolvido

A Figura 11 apresenta um diagrama ilustrativo do sistema desenvolvido. Na seqüência está exposta a descrição das principais funcionalidades de cada um dos componentes do diagrama.

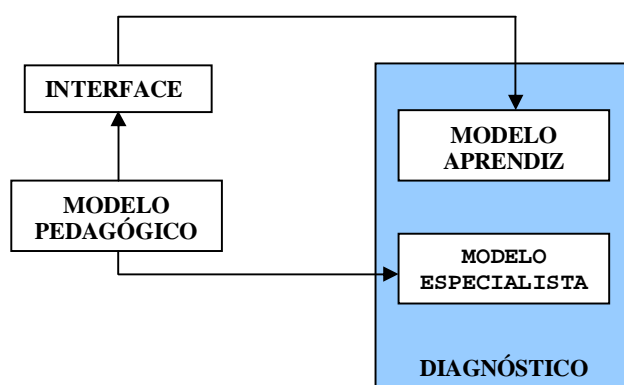


Figura 11 - Modelo do sistema desenvolvido

INTERFACE – Permite a interação do aprendiz com o sistema. Procurou-se desenvolver uma interface simples, de fácil utilização, intuitiva de forma a simplificar a navegação do usuário. Foram respeitados aspectos de interação homem computador no desenvolvimento da interface; na maneira como o usuário é conduzido no diálogo, fornecendo, por exemplo, *feedback*, durante a correção dos exercícios, a possibilidade de evitar e reduzir a ocorrência de erros do usuário; escolha de denominações significativas para os usuários e organização das informações apresentadas no sistema facilitando o estudo do conteúdo.

Considerou-se importante a facilidade de navegação pelo fato de que não seria interessante a interferência de fatores externos ao processo de aprendizagem. Assim o aprendiz centra-se no que está aprendendo e rapidamente vai navegando de forma a encontrar os conteúdos necessários à complementação do que está estudando. Sua atenção não é desviada por aspectos alheios ao processo em foco, como, por exemplo, ter que parar sua linha de pensamento para pensar como vai acessar algum outro item. Procurou-se atentar para o fato de tentar manter o foco de pensamento. Foi esta a idéia que levou a elaboração da interface.

MODELO APRENDIZ - O aprendiz poderá também escolher uma das lições para fazer seus exercícios e verificar como está seu conhecimento. Em relação à verificação dos exercícios, este modelo atua permitindo que o aprendiz aponte sua alternativa e em seguida solicita a verificação do mesmo por parte do sistema.

MODELO ESPECIALISTA – No modelo especialista está a “inteligência” do sistema. É o especialista que armazena os conceitos corretos. É o especialista que faz a correção dos exercícios de probabilidade e estatística apontados pelo aprendiz (Figura 11).

MODELO PEDAGÓGICO - Para a determinação do modelo pedagógico não houve estudo de teorias de aprendizagem e nem o acompanhamento de algum especialista na área. Para o sistema proposto utilizou-se o método de acompanhamento, para que o professor possa ter um acompanhamento do que os usuários estão fazendo no sistema, quais as lições já acessadas e quais exercícios já foram feitos, para permitir ao professor uma forma de acompanhamento, procurando eliminar ou pelo menos minimizar as dúvidas apresentadas por cada usuário.

4.2 Escopo do Trabalho

Agora apresentaremos um fluxograma para ilustrar o funcionamento do sistema na interface do aprendiz em seguida o fluxograma será explicado.

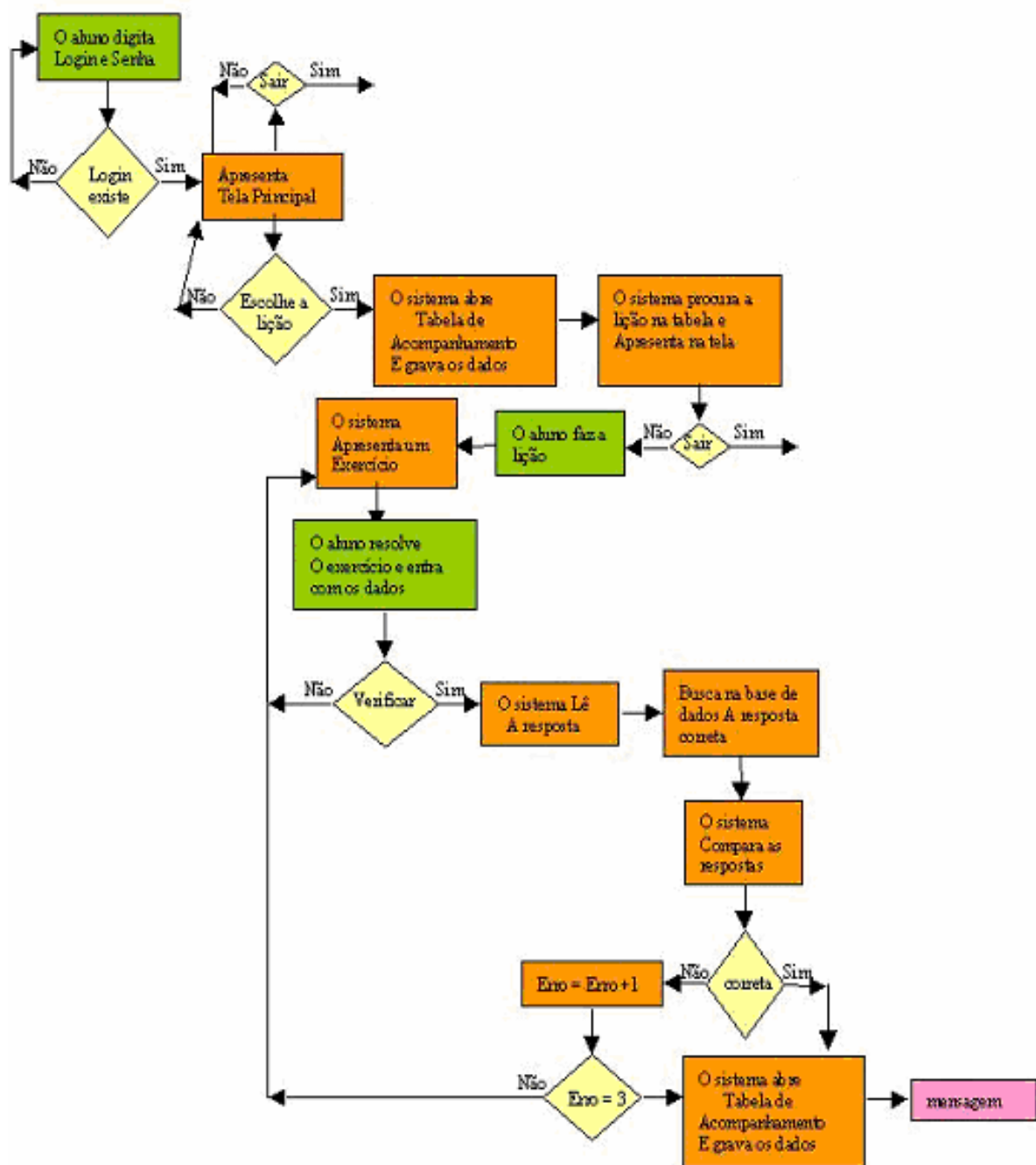


Figura 12 - Interação com o sistema

Para acessar o sistema é preciso digitar um login e uma senha que são cadastrados pelo professor, através de um cadastro existente no sistema, mas que só está disponível para o professor que passa a ser o administrador do sistema. As informações cadastrais do aprendiz são arquivadas em uma tabela existente no banco de dados.

Quando o login e a senha são inseridos pelo usuário o sistema vasculha a tabela para verificar se o registro existe, se o nome de usuário existe. Ele verifica então se a senha está correta e só então libera a entrada do aprendiz no sistema, esta implementação foi necessária para que o sistema pudesse reconhecer qual aprendiz está acessando o sistema para fazer o acompanhamento de seus passos dentro do sistema, estes dados serão utilizados para a emissão de um relatório ao professor, isso será melhor explicado posteriormente.

Ao entrar no sistema o aprendiz tem à sua disposição uma lista e um menu com todas as lições contidas no sistema. Este menu é fixo e estará sempre presente em todas as telas para que o aprendiz possa mudar de lição se necessário.

Quando o aprendiz escolhe uma das lições para acessar ele precisa dar um click de mouse em cima do link ou do menu. Ao fazer isso o sistema primeiramente abre uma tabela que contém as informações de acompanhamento do aprendiz e grava nela um registro contendo o nome do aprendiz, a data, a hora, e qual lição está sendo acessada. Logo após, abre uma outra tabela que contém todas as lições do sistema, encontra o registro da lição que está sendo acessada e exibe o conteúdo na tela do sistema tutor. Este procedimento é repetido todas as vezes que o aprendiz acessa uma lição.

Dentro das lições o aprendiz poderá avançar e retornar as páginas de acordo com sua necessidade. É possível também trocar de lição a qualquer momento, pois o menu contendo todas as lições é fixo na parte superior da tela. O sistema também permite que o aprendiz saia a qualquer momento da lição, apesar desse item aparecer apenas uma única vez no fluxograma.

Ao final de cada lição existe um exercício para ser resolvido pelo aprendiz. O exercício é apresentado para o aprendiz, que, após resolvê-lo, indica ou entra com a resposta em um campo próprio e solicita a verificação desta resposta, clicando no botão “verificar”.

Quando o botão “verificar” é clicado, o sistema busca em sua base de dados a resposta correta para o exercício e verifica se está igual à informada pelo aprendiz. No caso da resposta do aprendiz ser igual à resposta contida na base de dados, o sistema mostra uma mensagem ao usuário informando que sua resposta está correta e grava na tabela de acompanhamento o nome do aprendiz, a data, a hora, o número do exercício e o acerto. Mas, se a resposta não estiver correta, o sistema emite na tela uma mensagem ao aprendiz, informando que sua resposta não está correta e solicita uma nova tentativa.

Internamente o sistema inicializa um contador com o valor 1. Este contador tem como objetivo inibir a intenção do aprendiz de ficar tentando obter a resposta correta sem resolver o exercício. Na terceira tentativa incorreta o sistema emite uma mensagem solicitando ao aprendiz que procure resolver suas dúvidas sobre o assunto em questão e grava na tabela de acompanhamento, o nome do aprendiz, a data, a hora, o número do exercício e quantas vezes ele errou.

Ao terminar a lição e resolver o exercício, a lição é encerrada e esta informação também é gravada na tabela de acompanhamento, informando o nome, a data, a hora e qual lição foi finalizada pelo aprendiz. O fato de a lição estar finalizada pelo aprendiz não o impede de acessá-la novamente.

As informações que são gravadas na tabela de acompanhamento são utilizadas para a emissão de um relatório que está disponível para o professor. Este relatório é emitido por aprendiz e apresenta as informações de quais lições foram acessadas a data e hora que isso ocorreu, quais foram concluídas e quais ainda não estão concluídas, os exercícios resolvidos, os erros e os acertos.

Estas informações podem ser utilizadas pelo professor para conhecer quais as dificuldades do aprendiz e sua evolução no aprendizado, o que possibilita atuar sobre o problema ajudando o aprendiz em suas dúvidas.

O sistema do administrador possui todas as funções apresentadas anteriormente, mas inclui também duas particularidades que serão tratadas a seguir, resumidas na figura 13.

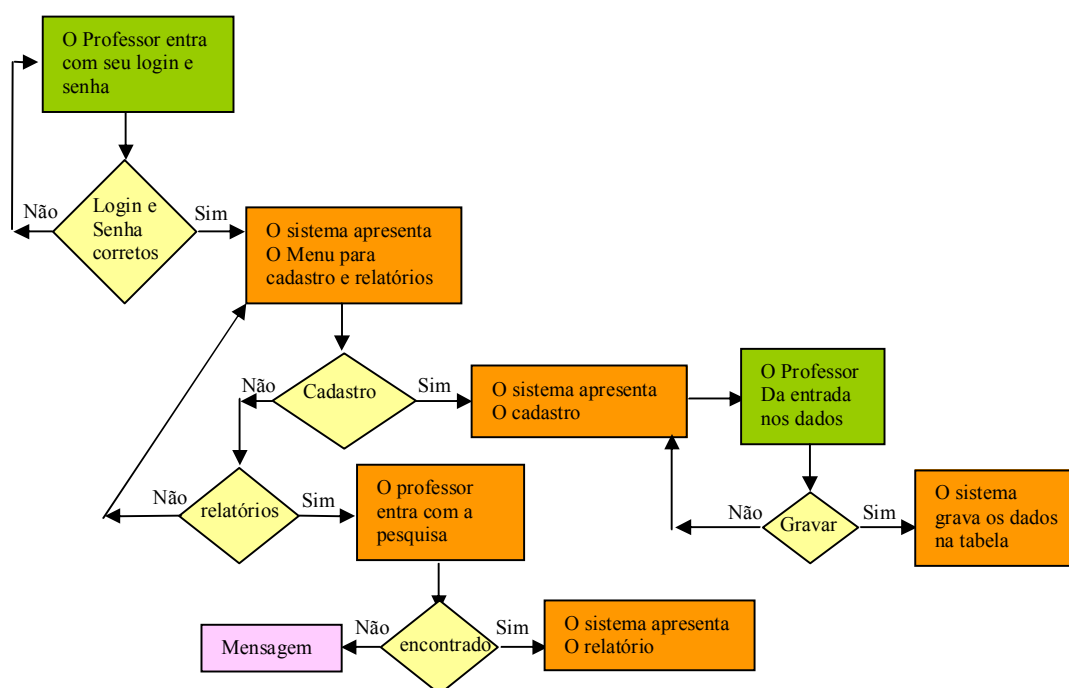


Figura 13 – Sistema do administrador

O sistema do administrador realiza o cadastramento dos aprendizes e a emissão dos relatórios de acompanhamento. Ao entrar com o login e com a senha o sistema procura em sua tabela se o login existe e se a senha está correta. Se estiver correto o sistema mostra na tela principal um menu com as características exclusivas do administrador: o cadastro e o relatório.

Se o professor optar por cadastrar um aprendiz, ao clicar na opção o sistema mostra na tela os campos que devem ser preenchidos para a inclusão do cadastro. O professor preenche os dados e solicita ao sistema sua gravação, internamente o sistema abre a tabela cadastro move os campos digitados para a tabela e grava o registro.

O sistema também permite ao professor alterar um cadastro ou excluir um existente, isso não foi mostrado no fluxograma para enxugar a estrutura e por possuir as mesmas características do processo de inclusão.

Quando o professor seleciona a opção relatório, o sistema mostra ao professor uma forma de pesquisa por aprendiz. Se o professor escolhe recuperar os dados sobre um determinado aprendiz, o sistema procura na tabela de cadastro este aprendiz. Se ele existir o sistema acessa seus registros de acompanhamento e emite o relatório.

O relatório apresenta as informações coletadas durante o acesso dos aprendizes ao sistema, onde está data, hora, itens acessados pelo aprendiz, exercícios realizados, erros e acertos. Com isso o professor poderá prestar uma assistência aos aprendizes que estiverem com dificuldades.

4.2.1 Telas do sistema.

Apresentaremos agora algumas telas do sistema.

Para iniciar a navegação no sistema o aprendiz deve estar cadastrado pelo professor, isso se faz necessário para que o sistema possa acompanhar os passos do aprendiz dentro do sistema para emitir o relatório de acompanhamento ao professor. Como mostra a figura 14.



Figura 14 - Login do sistema

Quando o aprendiz faz o login no sistema uma variável que o identifique é arquivada no banco de dados e cada lição acessada por este aprendiz dentro do sistema é gravada em uma tabela. Isso se faz necessário para que o sistema possa emitir um relatório de acompanhamento, por aprendiz, para o professor, quando solicitado. Esta opção de relatório esta disponível apenas na versão do administrador (professor), assim como o cadastramento dos aprendizes no sistema, conforme visto na figura 15.

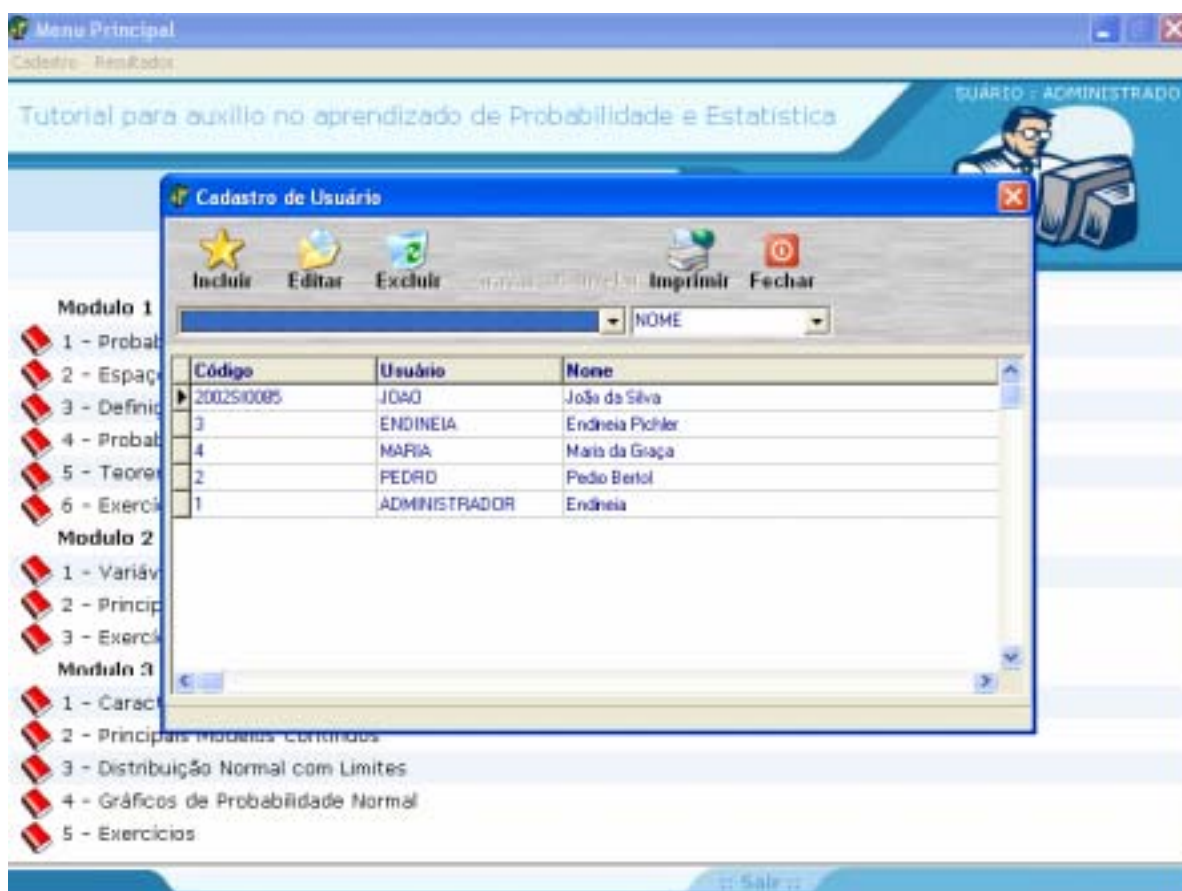


Figura 15 - Tela Principal administrador (professor)

O professor é o administrador do sistema. Somente ele pode efetuar o cadastramento de novos aprendizes, acessar e imprimir as informações do cadastro dos aprendizes já cadastrados no sistema, como pode ser visto na figura 16.

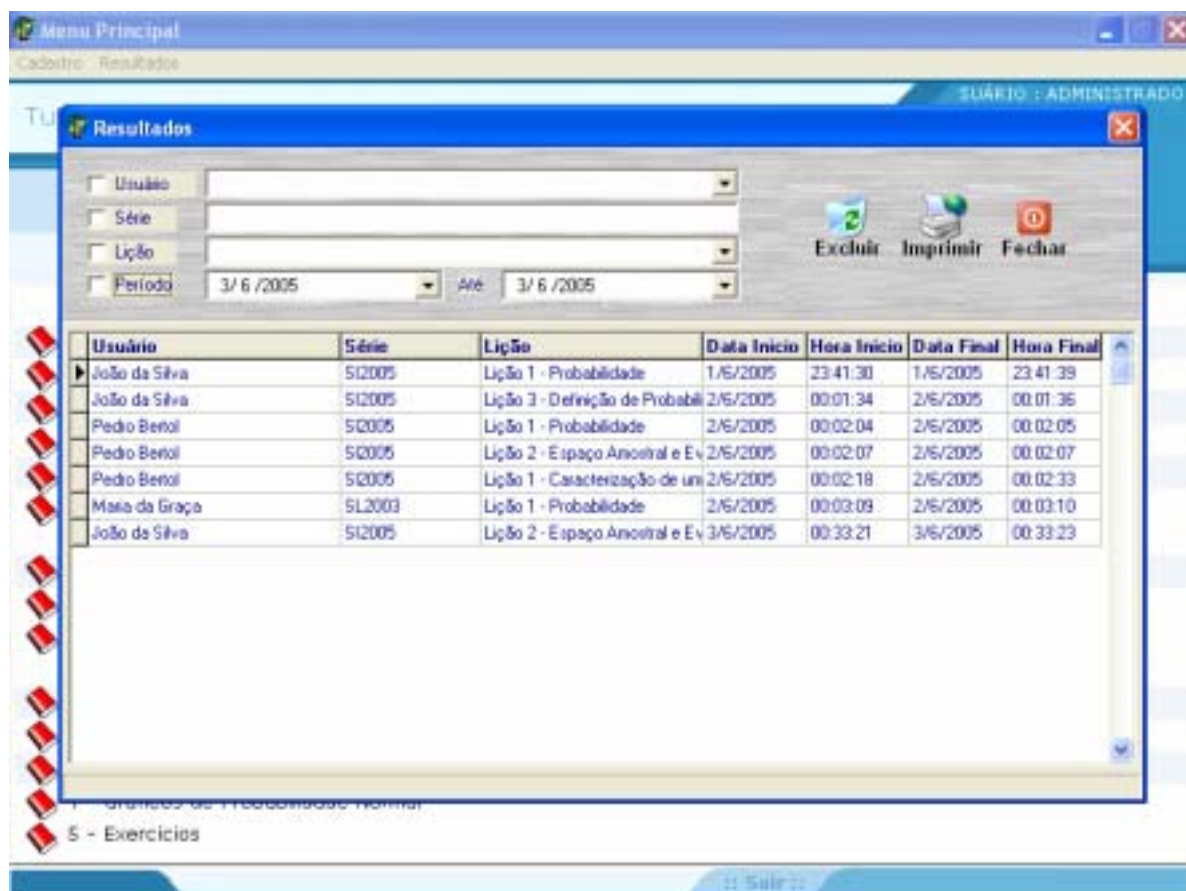
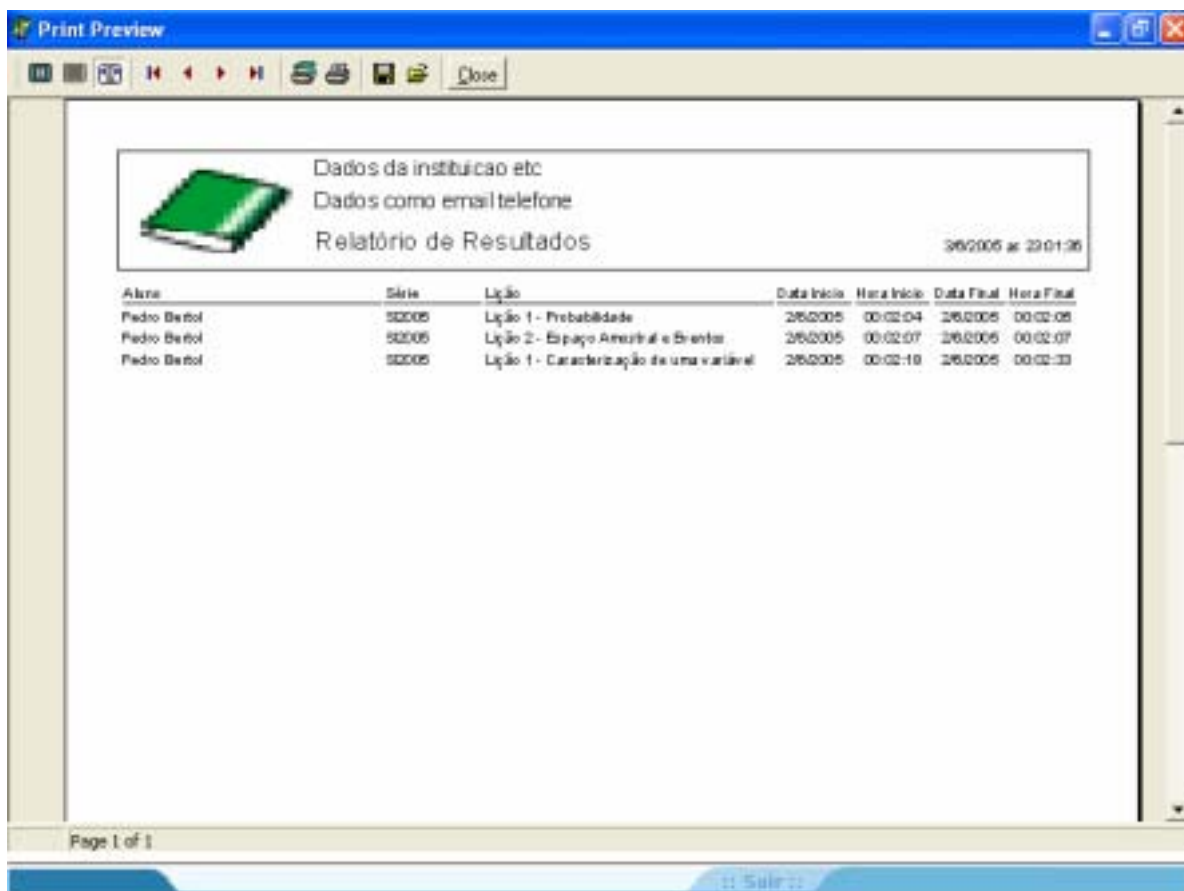


Figura 16 - Tela Principal administrador (professor)

Quando o professor solicita ao sistema o acompanhamento, para que ele possa fazer uma verificação através de um relatório do que os aprendizes estão fazendo no sistema, quais lições já foram acessadas, o sistema emite o relatório com as informações solicitadas pelo professor como mostra a figura 17.



Aluno	Série	Lição	Data Início	Hora Início	Data Final	Hora Final
Pedro Bertol	S2005	Lição 1 - Probabilidade	26/02/05	00:02:04	26/02/05	00:02:05
Pedro Bertol	S2005	Lição 2 - Espaço Amostral e Eventos	26/02/05	00:02:07	26/02/05	00:02:07
Pedro Bertol	S2005	Lição 1 - Caracterização de uma variável	26/02/05	00:02:10	26/02/05	00:02:33

Page 1 of 1

Figura 17 - Acompanhamento Professor

O relatório mostra o nome do aprendiz, sua série, a lição, a data e a hora em que o aluno iniciou a lição e a data e hora em que ele concluiu a lição.

O objetivo da emissão desse relatório é fazer com que o sistema possa trocar informações com o professor, tornando-se um auxiliar no processo de ensino-aprendizagem. Isto permite ao professor perceber as dificuldades do aprendiz e seu interesse pelo sistema, para ajudá-lo a dirimir, ou pelo menos minimizar, as dúvidas encontradas.

O aprendiz também pode acompanhar seu aprendizado através do sistema que mostra ao aluno quais lições ele já concluiu e quais faltam ser feitas.

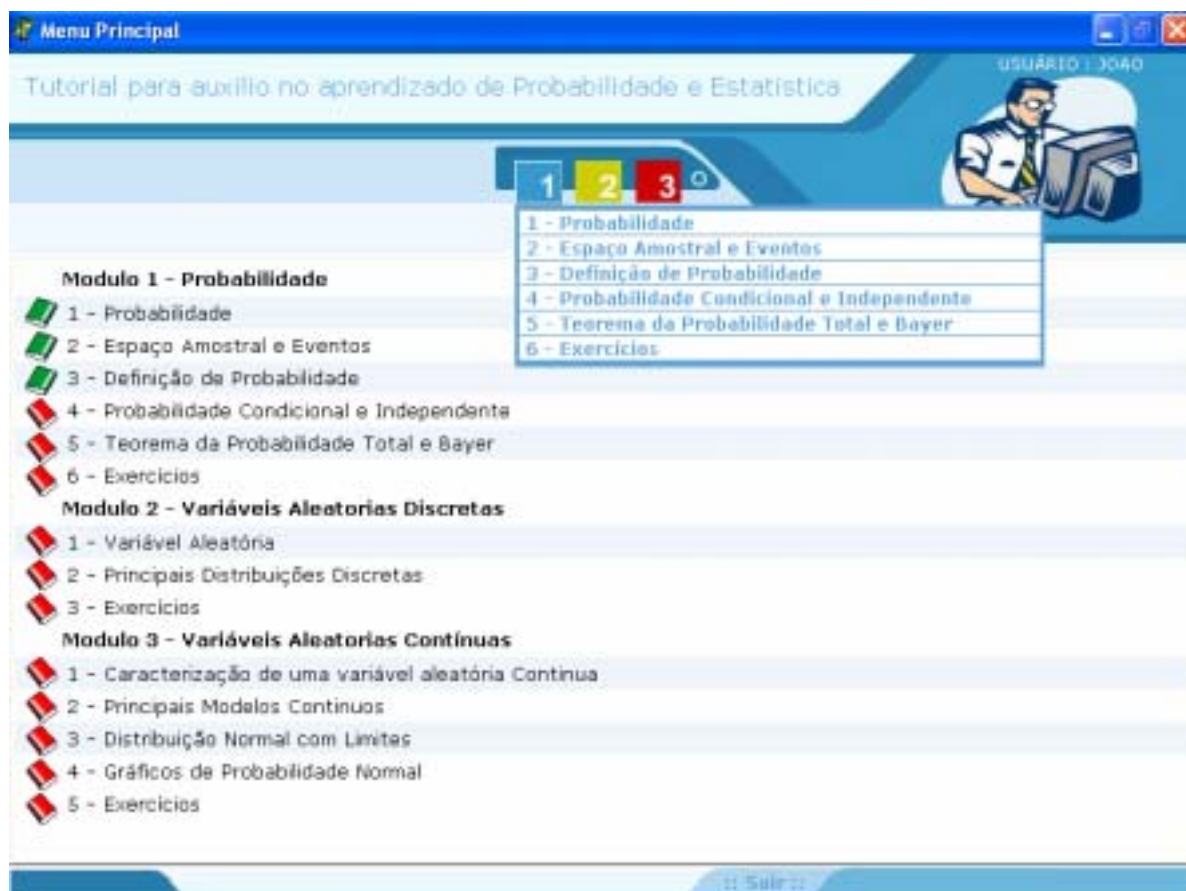


Figura 18 - Acompanhamento aluno

O acompanhamento é apresentado ao aprendiz, separado por lições, os ícones dos livros apresentados à frente do nome de cada lição representam o estado da lição no momento: quando a lição ainda não foi concluída pelo aprendiz o livro aparece fechado em vermelho, quando o aprendiz conclui a lição o livro aparece na cor verde e virado para baixo.

A Figura 18 apresenta as principais funcionalidades do sistema. Com o sistema Tutor é possível, consultar em uma base de dados conceitos de Probabilidade, variáveis aleatórias discretas e variáveis aleatórias contínuas, separadas em módulos. De acordo com a metodologia desenvolvida para este sistema, cada módulo é composto de algumas lições e ao final de cada modulo o sistema apresenta alguns exercícios para que o aprendiz fixe o que aprendeu até então.

Através da tela principal do sistema o aprendiz pode escolher uma das lições que deseja estudar. O aprendiz poderá acessar as lições tanto pelo menu quanto pelos links apresentados no centro da tela. Após escolhido o assunto a ser estudado, o aprendiz clica no nome da lição e ela é apresentada, como mostra a figura 19, que traz um exemplo de uma das lições que é apresentada pelo sistema. As Lições estão armazenadas em banco de dados.

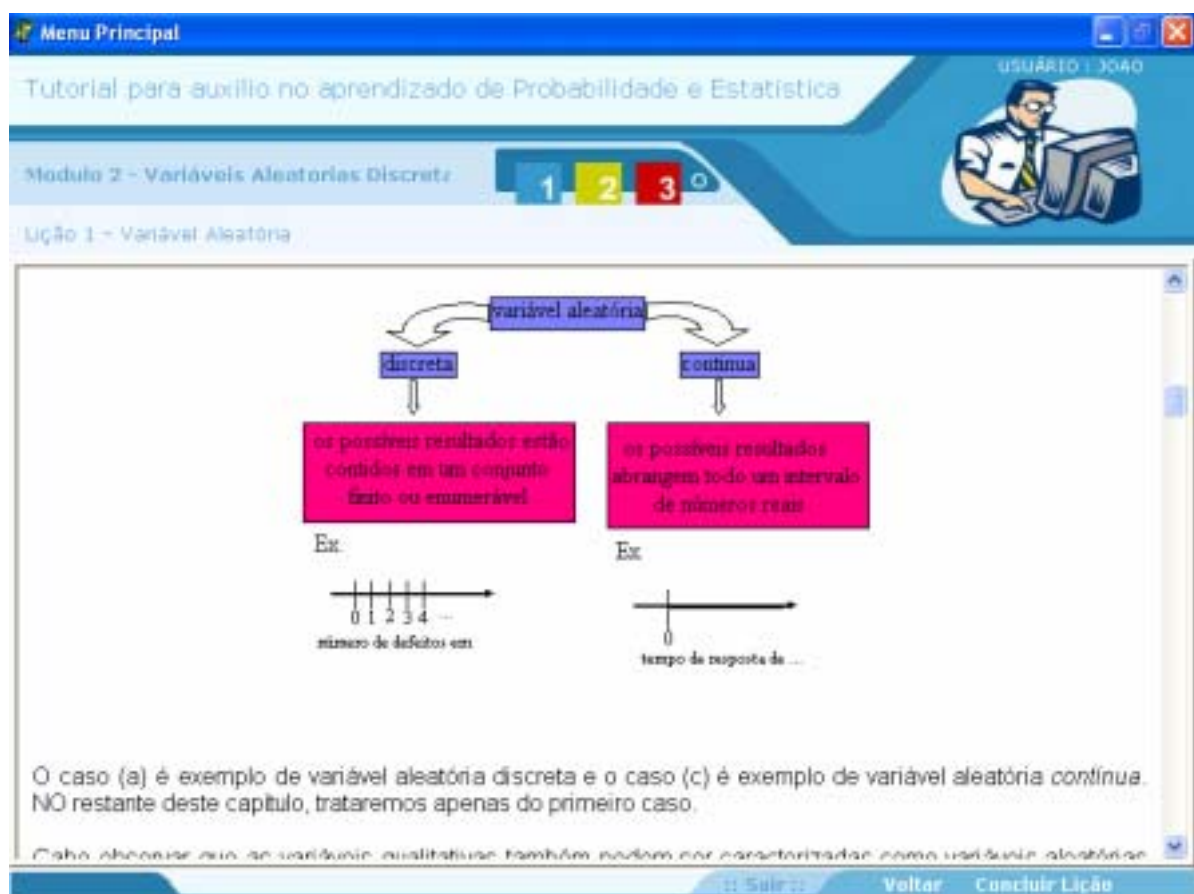


Figura 19 - Lições

O sistema faz a verificação dos exercícios nele contidos. A figura 21 traz um dos exercícios apresentados pelo sistema.

Menu Principal

Tutorial para auxílio no aprendizado de Probabilidade e Estatística

USUÁRIO: JOAO

Modulo 1 - Probabilidade

1 2 3

Exercício 1

1) Retira-se, ao acaso, uma carta de um baralho de 52 cartas. Calcule a probabilidade.

a) De carta ser de ouro. /

b) De uma carta ser de ouro ou uma figura. /

Verificar

2) Para testar se um sistema especialista responde satisfatoriamente as perguntas de um usuário, fez-se 5 perguntas de múltipla escolha com 4 alternativas de resposta. Se o sistema escolhe as alternativas aleatoriamente, qual é a probabilidade de ele responder corretamente todas as 5 perguntas.

Resposta /

Verificar

Sair Voltar Concluir Lição

Figura 20 – Exercício apresentado pelo sistema

A figura 22 mostra um caso onde o aprendiz respondeu e solicitou a verificação.

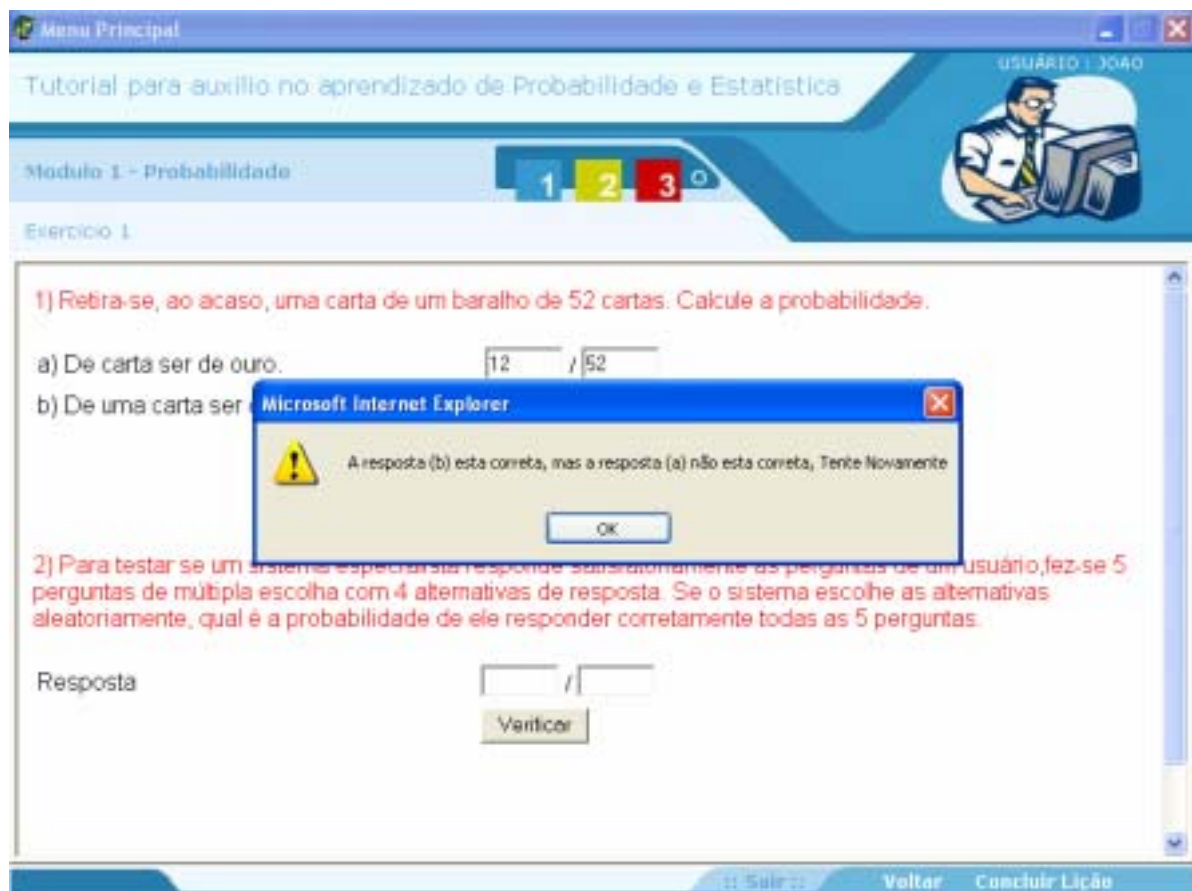


Figura 21 - Resultado da verificação

Tomou-se o cuidado com o teor e conteúdo das mensagens emitidas ao usuário de forma a não causar impacto negativo. O sistema objetiva ser um auxiliar no processo de ensino, assim a correção do que o aprendiz fez tem que levá-lo a querer aprender mais e sanar suas dificuldades.

4.3 – Considerações finais

Neste capítulo foram apresentadas as principais características do sistema tutorial inteligente desenvolvido para o auxílio ao ensino de Probabilidade. Foram mostradas as

formas como usuário (estudante) e professor (administrador) interagem com o sistema, e as várias funcionalidades embutidas no ambiente.

O sistema é limitado no que diz respeito a avaliação de funcionamento. O sistema não foi testado em sala de aula, tendo sido limitado a testes de funcionalidade realizados no final de sua elaboração.

No próximo capítulo serão apresentadas as conclusões, contribuições e sugestões para trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

Existe a necessidade de formas alternativas de ensino/aprendizagem, entretanto as maneiras de como organizá-las devem ser estudadas e discutidas. Este trabalho mostrou uma alternativa de apoio ao professor no processo de ensino da estatística e probabilidade.

A utilização de sistemas tutores inteligentes na educação pode facilitar a inclusão dos aprendizes no mundo digital, podendo ser um auxiliar na transmissão do conhecimento.

A pesquisa sobre a tecnologia de sistemas tutores inteligentes, revelou uma área de aplicação promissora. O interesse pela área é estimulado por uma série de mudanças ocorridas nas organizações, na sociedade e na própria tecnologia. A necessidade de ensinar/aprender de maneira alternativa e flexível é um dos fatores que incentivam o desenvolvimento de *softwares* para a educação.

O sistema proposto preconiza a utilização de *softwares* na educação, facilitando assim a forma de ensinar/aprender. Durante a elaboração do sistema foram escolhidos os conceitos que fariam parte do sistema e optou-se por apresentar os conceitos básicos de probabilidade, variáveis aleatórias discretas e variáveis aleatórias contínuas.

Inicialmente havia sido planejada a verificação de exercícios que pudessem ser elaborados pelo próprio aprendiz. Mas para que isso fosse possível seria necessário o desenvolvimento de uma interface adaptativa bem como uma lógica de programação que pudesse interpretar e resolver o exercício elaborado pelo aluno necessitando para tanto que tivesse em seu código técnicas mais complexas da inteligência artificial. Devido à complexidade deste tipo de desenvolvimento optou-se por tratar apenas de exercícios fixos do sistema. O sistema proposto apresenta o conteúdo e ao final traz exercícios do conteúdo para que o aprendiz perceba como está seu aprendizado. O aprendiz pode verificar se a resposta por ele encontrada está correta, solicitando ao sistema a verificação do exercício resolvido. Para o professor o sistema apresenta um relatório de acompanhamento, onde estão as lições acessadas

por determinado aprendiz, permitindo avaliar como ele está se saindo, em quais lições há mais dificuldades, entre outras informações.

A linguagem de programação utilizada, através do ambiente Borland Delphi®, apresentou recursos suficientes para o desenvolvimento pretendido, possibilitando a integração de novos conceitos quando necessário. No entanto nada impede que este sistema seja implementado em outras linguagens. O Delphi foi a linguagem de programação escolhida devido à sua fácil utilização, sua grande aceitação no mercado e por ter todos os recursos necessários para o desenvolvimento deste trabalho.

Como resultado deste trabalho obteve-se um sistema que pode contribuir no aprendizado de conceitos de probabilidade e estatística, no entanto muito ainda pode ser feito sobre este trabalho.

5.1 Perspectivas Futuras

Existem muitas possibilidades que podem ser implementadas nesse trabalho e como uma das perspectivas futura está a implementação de outros conceitos de probabilidade e estatística.

Outra perspectiva que pode ser implementada na sequência é a apresentação dos exercícios pelo próprio aprendiz. Dentro de um parâmetro estabelecido o aprendiz entraria com os dados, e ele e o sistema resolveriam o problema. O sistema então poderia tutorar o aprendiz, observando seu desempenho e mostrando seus acertos e erros. Tal abordagem tornaria o sistema mais flexível e mais abrangente.

Os dados armazenados poderiam servir para duas diretrizes de diagnóstico. A primeira, para acompanhar a evolução do próprio aprendiz e a segunda para avaliar a eficiência do modelo pedagógico. O sistema poderá construir uma base de dados com informações sobre

como o aprendiz reage ao conteúdo apresentado, em que momento o aprendiz teve maior dificuldade e onde teve maior facilidade, por exemplo. Posteriormente, o sistema poderia acessar esta base de dados e modificar a ordem dos conteúdos ou até mesmo a forma de apresentá-lo: o sistema estaria aprendendo com os próprios erros do aprendiz. Nesta fase, então haveria necessidade de acompanhamento e auxílio de um profissional da área abrangida pelo sistema para que fosse implementado um modelo pedagógico adequado aos objetivos de estudo e as peculiaridades de um sistema informatizado.

O que também poderá ser feito é a avaliação do sistema em sala de aula, para então perceber melhor suas deficiências.

5.2. Contribuições

Podemos enumerar as seguintes contribuições deste trabalho:

- Pesquisa sobre conceitos de Estatística, Probabilidade, e Sistemas Tutores Inteligentes.
- Delineamento das características que um sistema tutorial inteligente deve ter para auxiliar no ensino de Probabilidade e Estatística.
- Implementação de um protótipo de sistema tutorial inteligente para auxílio ao ensino de Probabilidade, de tal forma que é possível a futura expansão do sistema (com a inclusão de outros conceitos de Probabilidade) ou a sua adaptação para outras áreas de conhecimento.

O sistema aqui apresentado é uma forma alternativa de ensinar probabilidade e estatística. Fornecendo aos aprendizes o acesso às funcionalidades do sistema de forma interativa.

5.3 Conclusão

Os sistemas tutoriais inteligentes são apenas um dos tipos de *softwares* educacionais existentes, no entanto apresentam-se como os mais promissores atualmente como auxiliares em processos de aprendizagem. Um sistema tutor inteligente sabe o que ensinar, como ensinar, e apreende informações relevantes sobre o aprendiz que está sendo tutorado, proporcionando um aprendizado individualizado.

O sistema desenvolvido, ainda que bastante simples, possui implementados os quatro modelos (componentes) básicos de um sistema tutor inteligente, que são a interface (interação do aprendiz com o tutor), o modelo do especialista (que é domínio de conhecimento do sistema), o aprendiz (estudante) e modelo pedagógico (que é a forma de acompanhamento do aprendiz).

REFERÊNCIAS BIBLIOGRÁFICAS

- ALJOCYR P. **Estatística Fundamental**, 2ª edição revista. Porto alegre, Sulina, 1979.
- AUBERGER, M. **Student Modeling in Educational Multimedia Titles Using Agents**, disponível em <http://aif.wu-wien.ac.at/usr/geyers>.
- AMIT, M. Learning probability concepts through games. Proceedings of Fifth International Conference on teaching Statistics, Volume 1, pp. 45-47, Cingapura, Junho de 1998.
- AZEVEDO, B. F. T., **Tópicos em Construção de Software Educacional**, Estudo Dirigido, Mestrado em Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, 1997.
- BARBETTA, P. A. , Reis, M. M., BORNIA, A. C. **Estatística: para curso de Engenharia e Informática**. São Paulo: Atlas, 2004).
- BECK, J. Stern, M. Haugsjaa, E. **Applications of AI in Education**, ACM Crossroads Student Magazine, 1996.
- BOYLE, R.G. PACE 2000- **Stand alone software package and add-in for Microsoft Excel designed for teaching statistics at introductory-intermediate level**. Proceedings of Fifth International Conference on teaching Statistics, Volume 2, pp. 809-815, Cingapura, Junho de 1998.
- CLARKSON, D.B., DONELL, D., HUNT, E., MADIGAN, D., MINSTRELL, J., TRAYNOR, C. VITAL: **An Intelligent Tutorial System for Statistics**. Proceedings of the Section on Statistical Education. American Statistical Association, pp. 88-93, 1994.
- DAVIES, J. E. **Multimedia and Hipermedia as Mindtools**, MORGAN Kaufmann Publishers, Inc., 1987.
- FEIGENBAUM, E. A. e BARR, A. **Applications-Oriented, AI Research: Education. Em The Handbook of Artificial Intelligence - Capítulo IX**. Los Altos, CA. William Kaufmann, 1982.
- FEU C. Du, HUNT, N., LEES, R. **Learning probability concepts using Microsof Excel**. Proceedings of Fifth International Conference on teaching Statistics, Volume 2, pp. 855-861, Cingapura, Junho de 1998.
- FINZER, W., ERICKSON, T.E. DataSpace – **A Computer learning environment for data analysis and statistics based on dynamic dragging, visualization, simulation, and networked collaboration**. Proceedings of Fifth International Conference on teaching Statistics, Volume 2, pp. 825-829, Cingapura, Junho de 1998.
- GIARDINA, M. **Interactive Learningnng Environments**. Springer-Verlag, 1992.
- Giraffa, L.M.M.; Nunes, M. A.; Viccari, R.M. (1997) Multi-Ecological: an **Learning Environment using Multi-Agent architecture**. MASTA'97: Multi-Agent System: Theory and Applications. Proceedings.. Coimbra: DE-Universidade de Coimbra.
- GREER, J. E. McCalla, G. I. **Student Modelling: The Key to Individualized Knowledge-Based Instruction**, NATO ASI Series, vol. 125, 1991.
- HALFF, H. M. **Curriculum and Instruction in Automated Tutors**, em “Foundations of Intelligent Tutoring Systems” Eds. Polson, M. & Richardson, J. J. Lawrence Erlbaum Associates Publishers, New Jersey, 1988.
- KEARSLEY, G. **Artificial Intelligence and Instruction: Applications and Methods**. Addison-Wesley Publishing Company, Inc., 1987.

KEELER, C., STEINHORST, K. A New Approach to Learning Probability in the First Statistics Course. *Journal of Statistics Education* Volume 9, Number 3 (2001)

KOMMERS, P. Grabinger, S. Dunlap, J. **Hipermedia Learning Environments – Instructional Design and Integration**, Lawrence Erlbaum Associates, Inc., 1996.

LAKATOS, E.M; MARCONI, M.A. **Técnicas de pesquisa 2º Edição**. São Paulo: Atlas, 1990.

MCLAUGHLIN, C. Oliver, R. **Who Is In Control? Defining Interactive Learning Environments**. Edith Cowan University, 1995.

MARTINS, A. G. & DONAIRE, D. **Princípios de Estatística**, 4º edição, São Paulo, Atlas S.A., 1993.

MURRAY, T. **Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art**. *Journal of Artificial Intelligence and Education*, vol. 10, 1999.

NASSAR, Silvia M. **Sistema Estatístico Inteligente para Apoio a Pesquisas Médicas**. Florianópolis, Dezembro de 1995. Tese (Doutorado em Engenharia Elétrica)- Programa de Pós-Graduação em Engenharia Elétrica, UFSC.

Noal, Renato B., **Ambiente Colaborativo para Ensino de Estatística com o SESTAT**, dissertação do Curso de Ciências da Computação, Florianópolis, UFSC, fevereiro de 2002.

PAIVA, A. M. **About User and Learner Modeling: An Overview**, 1995, <http://www.cbl.leeds.ac.uk/~amp/personal.html>

RUSSEL, R.J. **Scratching the surface of Probability**. *Proceedings of Fifth International Conference on teaching Statistics*, Volume 2, pp. 751-756, Cingapura, Junho de 1998.

REIS, M. M. **Um Modelo para o ensino do Controle Estatístico da Qualidade**. Florianópolis, Brasil. Tese (Doutorado – Pós- Graduação em Engenharia de produção), Universidade Federal de Santa Catarina, 2001.

STERNFELD, L **Aprender Português e língua Estrangeira em ambiente de estudo sobre o Brasil**: a produção de um material. Dissertação de mestrado em lingüística aplicada. Campinas: Instituto de Estudos da linguagem da Unicamp, 184p. 1996

SCHWIER, M, Misanchuk, E. **Interactive Multimedia Instruction**, Educational Technology Publications, N. J., 1993.

TRURAN, K. **Is it luck, is it random or does the dice know?** *Proceedings of Fifth International Conference on teaching Statistics*, Volume 2, pp. 757-763, Cingapura, Junho de 1998.

URBAN-LURAIN, Mark. **Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology**, disponível em <http://aral.cse.msu.edu/Publications/ITS/its.htm>, 1996.

WENGER, E. (1987) - **Artificial Intelligence and Tutoring Systems**, Morgan Kaufmann Publishers, Inc. 1987.

WILSON, B. G. **Constructivist Learning Environment: Case studies in instructional design**. Educational Technology Publications, 1996.

WOOLF, B. **Intelligent Tutoring Systems: A Survey**, Morgan Kaufmann Publishers, Inc., 1988.

OHLSSON, S. **Learning from performance of errors**. *Psychological Review* 3 (2) p. 241-262 1996.

PARK, O. **Functional Characteristics of Intelligent Computer-Assisted Instruction: Intelligent Features**. *Educational Technology*, June 1988.

YAZDANI, M., **Artificial intelligence and education, volume one, "Learning environments and tutoring systems"**. Norwood, NJ, EUA, Ablex, 1987.

<http://www.acm.org/crossroads/xrds3-1/aied.html>, , acessado em agosto de 2004.

<http://www.ascilite.org.au/conferences/melbourne95/smtu/papers/mcloughlin.pdf>, , acessado em julho de 2004.

<http://www.auburn.edu/academic/education/eflt/hm.html>, , acessado em agosto de 2004.